

Grado Universitario en Ingeniería Informática
2017-2018

Trabajo Fin de Grado

Desarrollo de un videojuego de rol estratégico con LibGDX

Ernesto Gabriel González Chacón

Tutor/es

Daniel Borrajo Millán

Leganés, octubre 2018



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento - No Comercial - Sin Obra Derivada**

RESUMEN

La presente memoria explica el desarrollo de un videojuego de rol estratégico utilizando la librería de Java *LibGDX*. Los juegos digitales de rol estratégico son aquellos en los que el jugador tiene el deber de desarrollar los atributos de los personajes pensando en lo que pueden aportar a la historia del juego. Juegos como *Fire Emblem*, franquicia de Nintendo de este género, incluso pueden obligar a reiniciar después de varias horas si no se ha gestionado bien el crecimiento de los personajes.

Se hará una breve introducción al mundo de los videojuegos a través de su historia y su actualidad. Posteriormente se comparará este trabajo con otros de la misma índole, y se darán ejemplos de otros videojuegos que comparten el género.

Una vez terminado de describir el problema, se ahondará en el videojuego creado, *Overworld Tactics*. Se explicará cada una de las clases que lo componen y la forma de jugar. En resumen, se ha creado un videojuego del género RPG táctico, con tres niveles de dificultad y dos mapas. Para ganar se tienen que superar ambos mapas en un mismo nivel; y, para perder, los enemigos tienen que derrotar a las tres unidades del jugador.

Aproximadamente 276 horas de trabajo y más de mil líneas de código dieron como resultado un videojuego que es compatible con cualquier ordenador, siempre que tenga instalada la máquina virtual de Java en su versión 1.6 o superior. No es necesario ningún otro periférico mas que un ratón. Tanto el juego, como el código, serán liberados de forma gratuita con licencia que permitirá la redistribución y modificación, con la obligación de atribuir el mérito al autor original del mismo.

Palabras clave: Videojuego; RPG; Java; LibGDX

AGRADECIMIENTOS

Ante todo agradecer a la Universidad Carlos III de Madrid y todos sus departamentos que me han ayudado en estos años de carrera. A los profesores y en especial a Daniel Borrajo Millán, por aceptar ser mi tutor y por su apoyo en las tutorías.

También quiero agradecer a mis padres y a mi familia en general por todo su apoyo y las ayudas dadas. A Alba, por preocuparse siempre de que acabara este trabajo a tiempo y a mis compañeros de carrera, trabajo y Nintenderos.com por ayudarme con las pruebas y proporcionarme retroalimentación.

DEDICATORIA

A mis padres, por darme la oportunidad de estudiar esta carrera en esta universidad.

A Alba, por su apoyo incondicional.

ÍNDICE GENERAL

1. INTRODUCCIÓN.	1
2. ESTADO DEL ARTE.	3
2.1. Historia de los videojuegos	3
2.1.1. Primera generación.	4
2.1.2. Segunda generación	4
2.1.3. Tercera generación	4
2.1.4. Cuarta generación	5
2.1.5. Quinta generación	5
2.1.6. Sexta generación	5
2.1.7. Séptima generación	6
2.2. Videojuegos en la actualidad	6
2.2.1. Mercado móvil y de ordenadores personales	7
2.2.2. Videojuegos independientes.	7
2.2.3. Géneros de videojuegos	7
3. PLANTEAMIENTO DEL PROBLEMA	8
3.1. Juegos de Rol Estratégicos	8
3.2. Definición del Problema.	8
3.3. Otros trabajos.	10
3.4. Herramientas utilizadas	11
3.4.1. Udacity	11
3.4.2. IntelliJ IDEA	11
3.4.3. LibGDX.	11
3.4.4. Tiled.	12
3.4.5. Safari	12
3.4.6. Safari Books Online	12
3.4.7. Stack Overflow	12
3.4.8. Lucidchart	12
3.4.9. Overleaf v2	12

3.4.10. GitHub	13
3.4.11. Mendeley	13
3.4.12. Administración del tiempo.	13
3.4.13. Merlin Project Express	13
4. DISEÑO DEL VIDEOJUEGO.	14
4.1. Descripción del videojuego	14
4.2. Personajes.	15
4.2.1. Héroes.	15
4.2.2. Enemigos	16
4.3. Trama	19
4.4. Jugabilidad	19
4.4.1. Objetivos	19
4.4.2. Formas de Interacción	19
4.4.3. Sistema de Juego	20
4.4.4. Progresión de la dificultad.	21
4.4.5. Fin de la partida	21
4.5. Estilo artístico	21
4.6. Descripción técnica	22
5. RESULTADO	23
5.1. Sistema de Juego	23
5.2. Descripción de las clases	26
5.2.1. ProjectCastleGame.	28
5.2.2. Map	29
5.2.3. InputProcessorHelp	30
5.2.4. Unit	31
5.2.5. Hero	32
5.2.6. Enemy.	32
5.2.7. Assets	33
5.2.8. Constants	33
5.2.9. Enums	33
5.2.10. TextureTools	34

5.2.11. ActionMenu	34
5.2.12. CreditsScreen	34
5.2.13. DamageOverlay	35
5.2.14. GameOverScreen	35
5.2.15. GameplayScreen	35
5.2.16. Information	36
5.2.17. MainMenuScreen	36
5.2.18. TurnMessage	37
5.2.19. VictoryScreen	37
5.3. Versiones	37
5.3.1. Versión 0.1.0	37
5.3.2. Versión 0.1.1	38
5.3.3. Versión 0.2.0	38
5.3.4. Versión 0.3.0	38
5.3.5. Versión 0.4.0	38
5.3.6. Versión 1.0.0	39
5.3.7. Versión 1.0.1	39
5.4. Pruebas y retroalimentación	39
6. ENTORNO SOCIO-ECONÓMICO	45
6.1. Planificación	45
6.2. Costes	47
6.3. Impacto socio-económico	49
6.4. Marco Regulador	49
7. CONCLUSIONES	50
BIBLIOGRAFÍA	51

ÍNDICE DE FIGURAS

2.1	Captura videojuego OXO [4]	3
2.2	<i>Spacewar!</i> ejecutándose [5]	4
2.3	Top 10 de consolas vendidas según <i>VGChartz</i> [10]	6
3.1	Final Fantasy Tactics [18]	9
3.2	Fire Emblem [19]	9
3.3	Primera pantalla de <i>Overworld Tactics</i>	10
4.1	Personajes de <i>Overworld Tactics</i>	18
4.2	Pantalla Principal de <i>Overworld Tactics</i>	19
5.1	Pantalla de inicio del primer mapa en nivel <i>Easy</i>	23
5.2	Imagen luego de pulsar sobre un héroe	24
5.3	Movimiento del héroe seleccionado	25
5.4	Información del enemigo	25
5.5	Demostración de ataque. Ha quitado 3 de daño al enemigo	26
5.6	Diagrama de flujo de la toma de decisiones de los enemigos	27
5.7	Diagrama de clases de <i>Overworld Tactics</i>	28
5.8	Versión 0.1.0	38
5.9	Enemigos “trepano” por las paredes del castillo	39
5.10	Segundo mapa del juego, en dificultad fácil	40
5.11	Número de horas jugadas a la semana versus qué tan divertido considera el juego	42
5.12	Dificultad de <i>Overworld Tactics</i>	43
6.1	Gantt inicial del proyecto	45
6.2	Gantt real del proyecto	46

ÍNDICE DE TABLAS

5.1	Problemas encontrados y soluciones	41
5.2	Pruebas ejecutadas	44
6.1	Objetivos y descripción del diagrama de Gantt	46
6.2	Costes totales de los recursos humanos	47
6.3	Costes totales de los recursos materiales	48
6.4	Coste total del proyecto	48

1. INTRODUCCIÓN

El presente trabajo tiene como finalidad explicar el desarrollo de un videojuego de rol estratégico llamado *Overworld Tactics*.

La razón por la que se tomó la decisión de crear un videojuego como Trabajo de Fin de Grado viene dada por el continuo crecimiento de la industria de los juegos digitales. En el año 2017 los videojuegos ingresaron en España un total de 1359 millones de euros, lo que equivale a un crecimiento del 16,9 % con respecto al 2016 y superando a otras industrias importantes del entretenimiento, como el cine (con 597 millones de euros facturados) y la música (con 232 millones de euros facturados) [1].

El objetivo principal del trabajo es la creación de un videojuego de género *RPG* estratégico, o *SRPG* por sus siglas en inglés (*Strategy Role Playing Games*), para ordenador usando la librería de Java *LibGDX*. Los objetivos secundarios son: aprender a utilizar *LibGDX* para crear videojuegos para ordenador y desarrollar experiencia en el ámbito de la creación y diseño de videojuegos.

Con respecto a la memoria, está compuesta por una serie de apartados que hacen referencia a toda la información necesaria para entender el desarrollo y culminación del videojuego creado. Empieza explicando el estado del arte en el escenario de los videojuegos y el por qué es tan importante que se sigan creando, dando datos de la situación actual. También se hace una breve historia de los videojuegos para ubicarse en el contexto.

A continuación se plantea el problema. Se explica qué es un videojuego de rol estratégico, se hacen comparaciones con otros trabajos y se explican todas las herramientas utilizadas. Esto es seguido por el diseño de *Overworld Tactics*; es en este capítulo donde se explica todo lo relacionado al juego digital desarrollado en este trabajo. También se hace especial hincapié en que la metodología de diseño que se ha escogido para la realización de este trabajo es la GDD (*Game Design Document*), la cual no menciona requisitos de forma explícita, sino implícita dentro del documento, como se verá en este apartado.

A continuación viene el capítulo del resultado. En el cual se explica el videojuego obtenido, ya que en el anterior solo se habla del diseño, también se describen las clases que lo componen, se detallan las distintas versiones obtenidas y se muestran las pruebas realizadas y la retroalimentación obtenida a través de una encuesta de quienes lo jugaron.

Luego de la explicación detallada del resultado, se habla del entorno socio-económico. Es decir, la planificación del proyecto, tanto la inicial como la final y las razones de porqué se desvió; los costes humanos y materiales de la elaboración del videojuego así como de esta memoria; el impacto social y económico del mismo (apartado 6.3); y el marco legal que regula este trabajo (apartado 6.4).

Por último, se encuentran las conclusiones y líneas futuras del proyecto. También hay tres anexos: el primero es el manual de usuario del videojuego elaborado, el segundo es

la encuesta pasada a las personas que probaron el videojuego y el tercero es un extenso resumen en inglés de esta memoria.

2. ESTADO DEL ARTE

Para este Trabajo de Fin de Grado se ha desarrollado un videojuego de rol estratégico por turnos. Sin embargo, antes de pasar a explicar en qué consiste el juego, conviene revisar la evolución que han tenido los videojuegos estos últimos años y cómo la industria ha pasado a ser tan importante.

2.1. Historia de los videojuegos

Los videojuegos han avanzado mucho desde el primero conocido. Creado por Alexander Douglas en 1952 y titulado *OXO* [2]. Es una versión del juego “tres en raya” elaborado para la computadora británica *Electronic Delay Storage Automatic Calculator* (EDSAC), la cual todavía se encuentra operando en el *University Mathematical Laboratory* en la ciudad de Cambridge [3]. Se puede apreciar una captura del mismo en la figura 2.1.



Fig. 2.1. Captura videojuego OXO [4]

A pesar de que el primer videojuego se realizó en 1952, fue más una prueba de concepto, que un videojuego en sí [2]. Por lo que se suele conocer a otro juego electrónico como el primero de la historia, este es *Spacewar!* (figura 2.2) y fue desarrollado en el Instituto Tecnológico de Massachusetts por Steve Russell, Martin Graetz y Wayne Wiitanen [2].

Es importante resaltar que los dos juegos mencionados anteriormente nacieron en una época en la que los ordenadores solo eran utilizados para fines de negocios e investigación, por lo que esta etapa podría llamarse de forma poética “la prehistoria de los videojuegos”.

Con el nacimiento de *Pong*, publicado por Atari en 1976 junto con su consola homónima [2], fue cuando comenzó la primera generación de videojuegos.

Para no profundizar demasiado en la historia de los videojuegos, se contarán las características principales de cada generación, así como las consolas principales y juegos importantes. Toda esta información obtenida de [6].



Fig. 2.2. *Spacewar!* ejecutándose [5]

2.1.1. Primera generación

El inicio de “la historia de los videojuegos” es aquí. Con la *Magnavox Odyssey*, la *Atari Pong* y la *Coleco Telstar* como principales consolas y la primera consola portátil de videojuegos, publicada por Nintendo en el año 1977. Como juego estrella destacó el ya comentado anteriormente *Pong*.

2.1.2. Segunda generación

El buque insignia de esta generación fue la *Atari 2600*, con más de 30 millones de unidades vendidas. También forman parte de esta generación la *Intellivision*, la *Colecovision*, la *Atari 5200 SuperSystem* y el debut de SEGA con la *SG 1000*.

Cabe destacar que con el nacimiento del *Commodore 64*, capaz de ejecutar videojuegos y al mismo tiempo ser un ordenador personal; además de la saturación de cartuchos y juegos de poca calidad para las consolas dedicadas, empezó la decaída de los juegos en consolas. Hasta el punto de que se llegó a creer que las máquinas dedicadas a los mismos ya no tenían sentido [2].

Este pensamiento cambió con la llegada de la siguiente generación.

2.1.3. Tercera generación

Nintendo revitalizó el mercado de los videojuegos con el lanzamiento en Japón de la *Famicom* (nombre que proviene de *Family Computer*) en 1983 que, siguiendo el ejemplo del éxito de la *Commodore 64*, fue un ordenador “familiar”, así como una consola dedicada. De hecho, venía con dos mandos ya integrados en ella.

Sin embargo, no fue hasta 1985 que Nintendo publicó la *Nintendo Entertainment System* (NES) en occidente, la cual era como una *Famicom*, pero dedicada exclusivamente

a los juegos electrónicos. Debido a que la crisis que existía en el mercado de los videojuegos venía sobretodo por la saturación del mismo y la falta de control, Nintendo fijó un estándar para evitar juegos de mala calidad y limitó el número de juegos al año por desarrollador. [7]

En el terreno de las portátiles, también fue la empresa nipona quien destacó con la *Gameboy* y el juego *Tetris*. La narración en los videojuegos también fue muy importante para esta generación, con juegos insignia como *Super Mario Bros.*, *Zelda* o *Final Fantasy*.

2.1.4. Cuarta generación

Si la tercera generación se caracterizó por el dominio de Nintendo, la cuarta se caracterizó por el inicio de la llamada "Guerra de Consolas"[6], donde las principales contendientes fueron la *MegaDrive* de SEGA y la *Super Nintendo*.

2.1.5. Quinta generación

Lo que caracterizó a esta generación fue el salto a las tres dimensiones, con la Nintendo 64 y sus *Super Mario 64* y *The Legend of Zelda: Ocarina of Time* como juegos principales que demostraron la potencia de la consola [8]. También fue la incursión de Sony en la batalla con su *PlayStation*, así como Apple, que de la mano de la empresa japonesa Bandai, lanzó la *Apple Pippin*, la cual no llegó ni a las cincuenta mil unidades vendidas [6].

En el terreno portátil, Nintendo lanzó una revisión de su *Game Boy* pero con juegos a color, a la cual llamó *Game Boy Color*.

2.1.6. Sexta generación

La última generación en la que SEGA competería en el terreno *Hardware* con la *DreamCast*, que no llegó a las 10 millones de unidades y convertiría a SEGA en una empresa desarrolladora de *Software*. Sony, en cambio, viviría su mayor logro con la consola más vendida de la historia, como se puede ver en la figura 2.3, la *PlayStation 2*, que además de sus gráficos y catálogo infinito de juegos, venía con un lector de DVD incorporado. [9]

También se caracteriza esta generación por la incursión de otro gigante del “Valle del Silicio” a la industria de las consolas, Microsoft, que temía que Sony, con su *PlayStation 2* con DVD, entre otras características entrara tan fuerte en el mercado doméstico, que compitiera directamente con Windows [11]. Nintendo, por su parte, lanzó la *GameCube*, que vendió veintiún millones de unidades, sin embargo muy por debajo de las 156 millones de unidades que vendió la *PlayStation 2*.

Pos	Platform	North America	Europe	Japan	Rest of World	Global
1	<i>PlayStation 2 (PS2)</i>	53.65	55.28	23.18	25.57	157.68
2	<i>Nintendo DS (DS)</i>	57.39	52.07	33.01	12.43	154.90
3	<i>Game Boy (GB)</i>	43.18	40.05	32.47	2.99	118.69
4	<i>PlayStation (PS)</i>	38.94	36.91	19.36	9.04	104.25
5	<i>Wii (Wii)</i>	45.51	33.88	12.77	9.48	101.64
6	<i>PlayStation 3 (PS3)</i>	29.42	34.55	10.47	12.46	86.90
7	<i>Xbox 360 (X360)</i>	49.11	25.87	1.66	9.16	85.80
8	<i>Game Boy Advance (GBA)</i>	40.39	21.31	16.96	2.85	81.51
9	<i>PlayStation Portable (PSP)</i>	21.41	24.14	20.01	15.26	80.82
10	<i>PlayStation 4 (PS4)</i>	26.92	32.03	6.76	14.21	79.92

Fig. 2.3. Top 10 de consolas vendidas según *VGChartz* [10]

2.1.7. Séptima generación

La consola más destacada de este período fue, sin lugar a dudas, la *Nintendo Wii*, que decidió salirse de la competencia por la potencia y apostar por mandos revolucionarios y el control por movimiento, lo que hizo que desbancara “en ventas y popularidad al resto de consolas” [6] con más de 100 millones de unidades vendidas.

Por otro lado, la *Xbox 360* ganó la batalla a la *PlayStation 3* con 83 millones de unidades vendidas, contra las 80 millones de consolas que vendió Sony.

Es importante resaltar que tanto Sony como Microsoft desarrollaron funciones de control por movimiento en sus consolas para competir con la *Wii* (*PlayStation Move* y *Kinect* respectivamente) [12] sin embargo no la alcanzaron en ventas.

2.2. Videojuegos en la actualidad

Actualmente, en cuanto al mundo de las consolas, nos encontramos en la octava generación. Las tres principales contendientes son la *Nintendo Wii U*, la *PlayStation 4* y la *Xbox One*. Sin embargo, esta generación ha sido particular, puesto que, tanto Sony como Microsoft, han desarrollado revisiones de sus consolas (la *PS4 Pro* y la *Xbox One X*) con mayor potencia y gráficos que sus predecesoras [6].

Por otro lado, la *Wii U* fue un fracaso en ventas, así que Nintendo desarrolló una consola completamente nueva uniendo los mercados portátil y de sobremesa: *Nintendo Switch*. Consola que todavía no se conoce en qué generación se encuentra, puesto que podría ubicarse en la octava, o bien, podría ser la primera consola de la novena generación. Todo depende del rumbo que se tome de ahora en adelante en la industria.

2.2.1. Mercado móvil y de ordenadores personales

El apartado de la historia de los videojuegos se centró sobre todo en las generaciones de las consolas, sin embargo no se puede dejar pasar el crecimiento que ha supuesto en la industria de los videojuegos la llegada de los teléfonos móviles inteligentes y las altas prestaciones en los ordenadores.

Un ejemplo de lo anterior es que en el año 2017 los beneficios en los videojuegos en España alcanzaron los 1.359 millones de euros. Teniendo en cuenta que solo el 23 % de las personas juegan en consolas y que hay un 22 % que prefieren jugar en ordenador y otro 21 % que lo hacen en móvil [13], no se puede negar que ambas plataformas también son muy importantes en el sector.

2.2.2. Videojuegos independientes

Debido a que el resultado de este Trabajo de Fin de Grado coincide con la definición de videojuego independiente, conviene por tanto, hablar de ellos.

Según Andrew Williams en su libro *History of Digital Games* [14], el término “indie” nace en el verano del 2000 con el *Scratchware Manifesto*, el cual fue un ensayo publicado por un grupo anónimo de desarrolladores criticando el estado de la industria de los videojuegos. Señalaban, entre otras cosas, que las grandes empresas seleccionaban los géneros a desarrollar de forma segura y sin arriesgar con innovaciones, con el fin de generar riqueza.

El mismo Williams también comenta en su libro que los videojuegos independientes siguieron creciendo gracias al apoyo de grandes empresas como Valve, Nintendo, Microsoft y Sony. Las cuales permitieron que se publicasen juegos *indies* en sus tiendas e incluso ayudaron a financiar algunos de ellos.

Actualmente es la tienda Steam, creada por Valve, la que se convirtió en el mayor refugio de los juegos independientes. Posee una enorme biblioteca de juegos que a día de hoy alcanza los veintitrés mil juegos [15] y sigue creciendo sin parar.

2.2.3. Géneros de videojuegos

El juego digital creado para esta tesis es de género rol y subgénero estratégico por turnos. En el siguiente capítulo se detallará lo que esas palabras significan. Sin embargo, antes de profundizar en dicho género, conviene mencionar los que existen actualmente: acción, aventura, cartas, conducción, construcción, cooperación, *cyberpunk*, deportivos, disparos, educativos, estrategia, lucha, musical, no interactivos, *pinball*, plataformas, puzzle, recopilatorio, antología, *RPG* (juegos de rol), simuladores, supervivencia y de tablero [16].

3. PLANTEAMIENTO DEL PROBLEMA

En este Trabajo de Fin de Grado se pretende realizar un videojuego de rol estratégico, por lo que conviene definir qué significa.

3.1. Juegos de Rol Estratégicos

Los juegos de rol, o RPG (*Role Playing Games*) por sus siglas en inglés, son todos aquellos juegos –nótese que no se habla solo de los electrónicos– en los que se encarna un papel. Claro está, con esta definición la gran mayoría de los videojuegos formarían parte de esta categoría, por eso es mejor ahondar más en la definición y no limitarse a la simple traducción.

Craig Stern, en su extenso ensayo publicado en la página web *sinisterdesign.net* [17], busca dar una definición a los RPG y concluye que un videojuego es del género rol si el jugador tiene que desarrollar a uno o varios personajes persistentes a través de decisiones tomadas por él, y que estas tengan efecto directo sobre el juego.

Los RPG estratégicos, por su parte, son un subgénero que toma esa última definición de los juegos de rol y añade toques de estrategia, como que se gestiona una mayor cantidad de personajes a través de batallas, generalmente, por turnos sobre mapas con casillas.

A continuación se mencionarán dos juegos de este subgénero que sirvieron de inspiración para el trabajo realizado y sus similitudes con este:

- Final Fantasy Tactics: este juego es de la serie de Final Fantasy, un RPG japonés de corte clásico, sin embargo lo convirtieron en estratégico para esta entrega. Las mecánicas de este videojuego son parecidas a *Overworld Tactics* en que se gestiona a un grupo de héroes a lo largo de varios mapas. Sin embargo, la perspectiva es isométrica, como se puede ver en la figura 3.1.
- Fire Emblem: el primero de esta saga que salió de Japón fue la mayor inspiración de *Overworld Tactics*. El tipo de mapa también se ve desde arriba y en algunos niveles es de tipo *overworld*. La mayor diferencia es que en Fire Emblem se utiliza un sistema de niveles, mientras que en este proyecto, los atributos de los personajes suben directamente. Se puede ver una imagen del juego en la figura 3.2.

3.2. Definición del Problema

Ahora que ya es conocido el género del videojuego que se ha realizado para este trabajo, es posible hablar de él de una forma más técnica.



Fig. 3.1. Final Fantasy Tactics [18]



Fig. 3.2. Fire Emblem [19]

El juego desarrollado tiene por nombre *Overworld Tactics* y es un SRPG (siglas de *Strategy Role Playing Game*). La palabra *overworld* en el título está referida al tipo de mapa usado en el mismo, que se caracteriza por presentar a los personajes a una escala superior a la del resto de objetos, con el fin de crear una percepción de que es una gran batalla la que se está presenciando. Un ejemplo de este tipo de mapa es la figura 3.3.



Fig. 3.3. Primera pantalla de *Overworld Tactics*

Se puede apreciar claramente que la escala de las unidades no es la misma que la de las edificaciones o los bosques. Este tipo de mapa permite que un personaje pueda estar “dentro” de un edificio o en el interior de un bosque sin necesidad de cambiar la perspectiva del mundo. Es muy utilizado en los juegos de este subgénero.

3.3. Otros trabajos

Para la realización de esta memoria se han tomado en consideración otros dos trabajos: uno de Máster y otro de Grado. Ambos de la Universidad Carlos III de Madrid.

El primero, escrito por Francisco López, consiste en el desarrollo de un videojuego de rol centrado en la exploración de mazmorras utilizando la misma librería que en este trabajo, de la cual se hablará más adelante [20].

El Trabajo de Fin de Máster realizado por López es un RPG, al igual que este, pero no es del subgénero estratégico, sino de exploración de mazmorras. Él mismo comenta dentro de su memoria lo influenciado que se encuentra su trabajo a la saga de Atlus llamada *Etrian Odyssey*, que se centra en que el jugador tiene que investigar calabozos y dibujar el mapa. La única similitud de este trabajo con el de López, es que el crecimiento de los

personajes se basa en la toma de decisiones del jugador, lo cual enmarca a ambos juegos en el género rol.

El segundo, escrito por Hugo Huerta, es un juego multijugador para Android, también desarrollado en *LibGDX* [21]. Este Trabajo de Fin de Grado solo se asemeja al escrito por Huerta en que se desarrolló en *LibGDX*, puesto que el género es totalmente distinto y además está basado en la plataforma Android, en lugar de ordenador.

3.4. Herramientas utilizadas

A continuación se explicarán también todas las herramientas utilizadas para la realización de este Trabajo de Fin de Grado y cómo se usaron.

3.4.1. Udacity

La aplicación web de cursos online, *Udacity*, se utilizó antes de iniciar con la programación para refrescar el lenguaje Java y aprender sobre la librería en la que se desarrollaría, que es *LibGDX* y se hablará de ella más adelante.

En concreto se realizaron dos cursos en esta web, ambos teniendo como instructores a Peter Heinrich y Jeremy Silver. El primero se llama *2D Game Development with LibGDX*, en donde se explican las nociones básicas para trabajar con *LibGDX* [22]. El segundo curso tiene por nombre *How to Make a Platformer Using LibGDX* y va más allá que el primero, explicando el uso de texturas, animaciones, así como la interacción entre entidades dentro de los videojuegos; todo esto creando un juego del género plataformas [23].

3.4.2. IntelliJ IDEA

El entorno de desarrollo de JetBrains [24] fue utilizado para la programación del videojuego, se evaluaron opciones como *Android Studio* o *Eclipse*, pero al final se decantó por *IntelliJ IDEA* por ser más cómodo y fácil de usar que las otras opciones.

3.4.3. LibGDX

LibGDX es una librería desarrollada en Java, de código abierto, que permite crear juegos para las siguientes plataformas: “Windows, Linux, Mac OS X, Android (2.2+), BlackBerry, iOS, Java Applet, Javascript/WebGL para los navegadores: Chrome, Safari, Opera, Firefox, Internet Explorer vía Google Chrome Frame” [25].

En el caso de este trabajo, solo es compatible con ordenadores que tengan instalada la Máquina Virtual de Java a partir de la versión 1.6.

3.4.4. Tiled

A pesar de que *LibGDX* es una librería muy completa, siempre es necesario un editor de niveles para crearlos de forma gráfica e importarlos. La decisión de usar *Tiled* [26] vino dada por la facilidad de importar los mapas y la información de los mismos al código fuente.

3.4.5. Safari

El navegador de Apple [27] fue el utilizado para investigar los recursos, tanto para el desarrollo del videojuego, como para la posterior memoria. También fue el escogido para usar las herramientas online de las que se hablarán a continuación.

3.4.6. Safari Books Online

Safari Books Online es una plataforma en la que se puede encontrar una gran cantidad de libros y cursos especializados [28]. Muchas de las referencias bibliográficas de este trabajo fueron consultadas allí.

3.4.7. Stack Overflow

La página web, gestionada por Stack Exchange Inc., *Stack Overflow* [29] fue una de las más visitadas cada vez que se necesitaba información técnica sobre *LibGDX*. Al ser la comunidad de desarrolladores más grande, siempre se conseguían respuestas.

3.4.8. Lucidchart

La herramienta web *Lucidchart* está especializada en todo tipo de diagramas, desde flujo hasta mapas mentales [30]. En este trabajo fue utilizada para el diagrama de flujo donde se explica cómo funciona la toma de decisiones de los enemigos, es la figura 5.6

3.4.9. Overleaf v2

Este escrito se ha hecho con \LaTeX utilizando la herramienta online *Overleaf* v2, [31] que ha surgido de la unión entre *ShareLaTeX* y *Overleaf*. Se ha utilizado esta herramienta por su facilidad para compilar código de \LaTeX y su compatibilidad tanto con *GitHub* como con *Mendeley*, de las cuales se hablará a continuación.

3.4.10. GitHub

GitHub es el repositorio más grande actualmente en internet de proyectos de programación [32]. Para el control de versiones de *Overworld Tactics* se utilizó *GitHub*, así como para el control de versiones del código fuente en \LaTeX de este trabajo escrito.

3.4.11. Mendeley

Para organizar las referencias bibliográficas se usó *Mendeley* en su versión para Mac [33]. La razón principal del uso de esta aplicación es que está integrada en *Overleaf v2*, lo que hace que sea muy sencillo crear la bibliografía.

3.4.12. Administración del tiempo

Este apartado difiere un poco del resto, puesto que para la administración del tiempo se utilizaron dos herramientas en lugar de una.

Timing para Mac [34] fue la primera que se utilizó en los inicios de la programación del videojuego, que luego fue cambiada por *Hours* para iOS [35] por la mayor comodidad que ofrece para empezar los contadores y su simplicidad con respecto a *Timing*, que tiene más funciones de las necesarias para este proyecto.

3.4.13. Merlin Project Express

Merlin Project Express es una aplicación de gestión de proyectos, con especial atención en diagramas de Gantt y presupuestos [36]. Más adelante en el capítulo 6 se verá el resultado del uso de esta herramienta.

4. DISEÑO DEL VIDEOJUEGO

La metodología escogida para el diseño y posterior desarrollo de *Overworld Tactics* es la GDD, *Game Design Document* por sus siglas en inglés, que traducido sería Documento de Diseño de Videojuego. Se utilizó como referencia el *post* de Leandro González en la página web especializada en la industria de los videojuegos *Gamasutra*, en el cual explica cómo crear un Documento de Diseño de Videojuego [37].

El Documento de Diseño de Videojuego no tiene requisitos de forma explícita sin embargo, como comentan John Sharp y Colleen Macklin en el capítulo *Game Design Documentation*, equivalen a los requisitos convencionales, puesto que estos muestran como un software va a funcionar así como sus objetivos [38]. Por lo tanto, todo lo que se especifique en este apartado de diseño se tiene que tomar como requisitos a cumplir en el videojuego.

Además de esta metodología, se evaluaron otras como Métrica, que fue descartada por lo complicada que es para el proyecto en cuestión; también se evaluaron métodos ágiles como el *SCRUM* sin embargo, al estar tan relacionado a los trabajos en equipo, no se pudo implantar de una forma correcta.

Al final la decisión de utilizar GDD vino por su especialización en videojuegos y lo completo que es al especificar cada característica de los mismos. Empieza desde una simple descripción del juego, se explican todos los personajes, así como la historia y el estilo artístico.

Además, se explicarán las dos partes más importantes de un videojuego: su jugabilidad y la descripción técnica del mismo. La primera refleja al juego en sí, con todas las funciones que tendrá y la segunda, detalles técnicos del mismo.

4.1. Descripción del videojuego

Overworld Tactics es un juego de rol estratégico por turnos con dos mapas y tres niveles de dificultad. Está pensado para un público casual que disfrute de la estrategia por turnos.

Las mecánicas de juego se centran en mover a los tres personajes que el jugador puede controlar y eliminar a todos los enemigos que se encuentran en la pantalla, situando a los héroes al lado de estos y atacándoles.

4.2. Personajes

Existen tres personajes que controla el jugador, a los cuales llamaremos héroes de aquí en adelante, y dos tipos de enemigos. Las texturas de los personajes fueron extraídas de [39] y poseen licencia CC BY 3.0.

Todos los personajes poseen tres atributos: ataque, defensa y salud. Cuando un ataque es realizado, el daño que genera el atacante es la resta entre el ataque del mismo y la defensa del rival; luego ese número se resta a la salud de la unidad defensora. Más adelante en el apartado 5.1 se explica con mayor detalle.

También se explicará la posición inicial de cada personaje en el mapa usando un sistema de coordenadas X, Y en el que el punto de origen será la celda superior izquierda. Como los enemigos pueden repetirse en el mismo mapa, se pondrán todas sus posiciones por mapa separadas por punto y coma (;).

4.2.1. Héroes

Los atributos de los héroes no cambian con la dificultad, los de los enemigos sí. Esto se explicará más adelante en el apartado 4.4.4.

- **Eirika:** es la protagonista del juego, princesa y heredera del reino. Es la que posee los atributos más balanceados entre los héroes.

Atributos primer mapa:

- **Ataque:** 8
- **Defensa:** 4
- **Salud:** 10
- **Posición:** 7, 15

Atributos segundo mapa:

- **Ataque:** 9
- **Defensa:** 7
- **Salud:** 12
- **Posición:** 3, 19

- **Christian:** es el hermano menor de Eirika. Tiene los atributos un poco más bajos que los de su hermana. No destaca ninguno en particular.

Atributos primer mapa:

- **Ataque:** 7
- **Defensa:** 3

- **Salud:** 9
- **Posición:** 9, 15

Atributos segundo mapa:

- **Ataque:** 10
- **Defensa:** 5
- **Salud:** 9
- **Posición:** 2, 19

- **Airmanagild:** es el coronel de mayor confianza. Posee la defensa y la vida más bajas de los héroes, pero un ataque muy alto.

Atributos primer mapa:

- **Ataque:** 10
- **Defensa:** 2
- **Salud:** 7
- **Posición:** 5, 15

Atributos segundo mapa:

- **Ataque:** 11
- **Defensa:** 2
- **Salud:** 9
- **Posición:** 1, 19

4.2.2. Enemigos

- **Esqueletos:** tienen un ataque alto, pero una defensa muy baja y poca salud.

- **Easy**

Atributos primer mapa:

- **Ataque:** 7
- **Defensa:** 0
- **Salud:** 5
- **Posiciones:** 2, 5; 9, 5; 6, 7

Atributos segundo mapa:

- **Ataque:** 8
- **Defensa:** 2
- **Salud:** 5

- **Posiciones:** 5, 8; 3, 11; 16, 10; 10, 6; 11, 10

- **Medium**

Atributos primer mapa:

- **Ataque:** 8
- **Defensa:** 1
- **Salud:** 5
- **Posiciones:** 2, 5; 9, 5; 6, 7

Atributos segundo mapa:

- **Ataque:** 9
- **Defensa:** 2
- **Salud:** 7
- **Posiciones:** 5, 8; 3, 11; 16, 10; 10, 6; 11, 10

- **Hard**

Atributos primer mapa:

- **Ataque:** 10
- **Defensa:** 3
- **Salud:** 6
- **Posiciones:** 2, 5; 9, 5; 6, 7

Atributos segundo mapa:

- **Ataque:** 10
- **Defensa:** 3
- **Salud:** 7
- **Posiciones:** 5, 0; 6, 17; 18, 7; 12, 8

- **Limos:** tienen los atributos más equilibrados que los esqueletos. Ataque y defensa medios, pero buena salud.

- **Easy**

Atributos primer mapa:

- **Ataque:** 4
- **Defensa:** 4
- **Salud:** 6
- **Posiciones:** 13, 8

Atributos segundo mapa:

- **Ataque:** 5
- **Defensa:** 5

- **Salud:** 7
- **Posiciones:** 13, 2

- **Medium**

Atributos primer mapa:

- **Ataque:** 6
- **Defensa:** 6
- **Salud:** 6
- **Posiciones:** 13, 8

Atributos segundo mapa:

- **Ataque:** 6
- **Defensa:** 5
- **Salud:** 7
- **Posiciones:** 13, 2

- **Hard**

Atributos primer mapa:

- **Ataque:** 7
- **Defensa:** 5
- **Salud:** 5
- **Posiciones:** 13, 8; 15, 14

Atributos segundo mapa:

- **Ataque:** 8
- **Defensa:** 5
- **Salud:** 5
- **Posiciones:** 13, 2

Los personajes pueden verse en la figura 4.1. Empezando por la izquierda en la primera fila están Airmanagild, Christian, Eirika; y los enemigos Esqueleto y Limo se encuentran en la segunda fila.



Fig. 4.1. Personajes de *Overworld Tactics*

4.3. Trama

El juego no tiene historia explícita. En la pantalla principal (figura 4.2) se puede escoger directamente entre los tres niveles de dificultad y empieza el juego a continuación en la primera pantalla.

Sin embargo, existe una historia implícita: El reino al que pertenecen los héroes ha sido invadido por esqueletos y limos, con lo que la población está aterrorizada y es deber del jugador derrotarles.

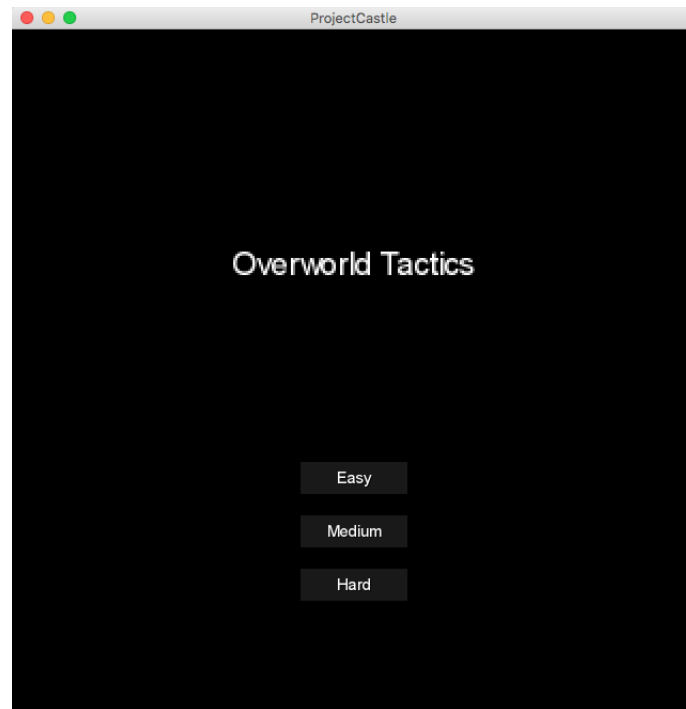


Fig. 4.2. Pantalla Principal de *Overworld Tactics*

4.4. Jugabilidad

4.4.1. Objetivos

El objetivo del juego es derrotar a todos los enemigos de ambas pantallas, para así llegar a la pantalla de victoria.

4.4.2. Formas de Interacción

Overworld Tactics se juega enteramente utilizando el ratón.

4.4.3. Sistema de Juego

El juego empieza con una pantalla de selección de dificultad. Una vez es escogida aparecerá el primer mapa con los enemigos y héroes ubicados en sus posiciones respectivas y con un mensaje de turno que aparecerá cada vez que el jugador o el ordenador acabe de realizar sus acciones.

Las acciones dentro de un turno se ejecutarán a través de un menú de acciones (en el caso de la CPU, se harán automáticamente siguiendo la toma de decisiones que se explicará más adelante). El susodicho menú tendrá dos botones: *Attack* y *Move*. En cada turno se podrán mover todos los personajes una única vez y atacarán también una vez. Si el ataque es lo primero que se realiza, luego no se podrá mover al héroe.

Para que aparezca el menú de acciones, se tiene que pulsar sobre un héroe que todavía no haya atacado en el turno o que no haya movido y haberse quedado lejos de los enemigos.

Para mover al personaje es necesario que este no lo haya hecho previamente en el turno. Pulsar sobre él, tocar el botón de *Move* y aparecerán sobre el mapa las casillas a las que puede moverse (el agua no es una de ellas). Para ir a una de ellas es necesario tocar sobre la misma. En caso de que el jugador no quiera ni mover ni atacar con el personaje, bastará con pulsar fuera del área de movimiento y ya quedará registrado. Si el movimiento acaba en casillas que no sean adyacentes a ningún enemigo, el héroe ya no podrá realizar más acciones en el turno, por lo que no aparecerá más el menú en caso de que se toque sobre él.

Para atacar es estrictamente necesario ubicarse en una casilla adyacente a un enemigo (las diagonales no cuentan), pulsar sobre el héroe con el que se desea atacar, luego pulsar sobre el botón *Attack* y por último tocar al enemigo al que se quiere atacar. El héroe ya no podrá realizar más acciones en el turno, por lo que no aparecerá más el menú en caso de que se toque sobre él. Los ataques tendrán una posibilidad del 85 % de ser realizados con éxito, sobre la unidad atacada saldrá en rojo la salud quitada o un guion negro si se ha fallado.

El turno acabará cuando todos los héroes acaben todas sus acciones. Aparecerá la ventana de cambio de turno y pasará al de los enemigos.

Además del menú de acciones, existe otra ventana dentro del juego que aparecerá al tocar sobre una unidad. esto es el menú de información, en el cual se desplegará el nombre de la unidad seleccionada y sus atributos de ataque, defensa y salud actualizados al momento. En el caso de los enemigos también aparecerá esta ventana al tocarles, junto con las casillas a las que se pueden mover en el mapa.

El desarrollo de los personajes se basa en la subida de los atributos cada vez que existe un ataque, esto es:

Para subir el ataque de un personaje, se tiene que luchar con él. La regla que existe es

que se compara el ataque del héroe con el del enemigo al cual se enfrenta. Si el enemigo tiene un ataque inferior en más de cinco puntos al de la unidad, entonces sube un solo punto; en el caso en que el ataque del enemigo sea superior en más de cinco puntos, sube tres puntos el del héroe; si no se cumplen ninguna de estas condiciones, el ataque sube dos puntos. Ocurre igual con la defensa, con la diferencia de que lo que sube es el atributo de defensa de la unidad que recibe el daño y se comparan estos puntos, en lugar de los de ataque.

Siempre que una unidad ataca, su ataque sube y la defensa de quien lo recibió también subirá. Los enemigos también se ven afectados por estas reglas, además, si el ataque es erróneo, eso no influye al aumento de estadísticas.

La toma decisiones de los enemigos vendrá basada en los atributos de los héroes. Atacarán a los héroes que puedan matar en un solo golpe, o bien, al que tenga la defensa más baja. Buscan primero dentro de su rango de ataque y si no hay ninguno, se acercan al que cumpla con las condiciones anteriores.

4.4.4. Progresión de la dificultad

Normalmente los *stats* de los enemigos no suelen subir de forma proporcional a los de los héroes en los RPG. Sin embargo, en este juego cada vez que se ataca a un enemigo, este sube su defensa, y cada vez que ataca a un héroe, sube su ataque. Por lo que en la misma pantalla los enemigos se pueden hacer fuertes rápidamente.

Otra cuestión a destacar es que la segunda pantalla tiene enemigos más fuertes que en la primera. En cuanto a los atributos, tanto los de los héroes como los de los enemigos mejoran, por lo que es más complicada de forma proporcional.

También existen tres niveles de dificultad distintos a elegir. Es importante destacar que los atributos de los héroes son los mismos en estos tres niveles, por lo que para ganar hay que basarse en la estrategia y en la suerte.

4.4.5. Fin de la partida

La única condición que se tiene que dar para perder la partida es que todos los héroes caigan derrotados. En ese caso aparecerá la pantalla de *Game Over* y se tendrá que reiniciar desde la pantalla de selección de dificultad.

4.5. Estilo artístico

El juego es 2D y con perspectiva cenital. El estilo de mapa es *overworld*, como se explicó anteriormente. A continuación se detallarán las fuentes de las que se extrajo el arte:

- Personajes [39]
- Texturas del mapa [40]
- Elementos de la interfaz de usuario [41]

4.6. Descripción técnica

La ventana del juego será de tamaño fijo de 640x640 píxeles para que quepan perfectamente ambos mapas de 20x20 casillas de 32 píxeles cada una. El número de fotogramas por segundo recomendado es de 60, sin embargo las bajadas no serán muy importantes al no ser un juego de acción rápida, sino de ritmo lento.

Estará programado en Java utilizando la librería *LibGDX*, la cual da facilidad de publicar el juego en varias plataformas aunque en principio solo será compatible con ordenadores. A través del entorno de desarrollo *IntelliJ IDEA* y el editor de niveles a utilizar será *Tiled* por su facilidad de exportación a la librería seleccionada.

5. RESULTADO

Es en este capítulo donde se enseñarán los resultados obtenidos siguiendo las guías planteadas en el capítulo anterior. Empezando por una descripción del sistema de juego final y luego embarcándose en apartados más técnicos donde se hablará del código del mismo.

5.1. Sistema de Juego

Este apartado es el resultado obtenido del apartado del mismo nombre del capítulo de diseño de esta memoria.

El juego empieza con una pantalla de selección de dificultad, como se puede apreciar en la figura 4.2. Los *stats* (atributos) de los héroes no cambian con la dificultad, pero sí los de los enemigos, lo que hace que se tenga que cambiar la estrategia al enfrentar cada nivel. Los mapas no varían según la dificultad pero el número de enemigos sí que puede variar.

Una vez el jugador selecciona la dificultad, pasa a la primera pantalla, que se puede apreciar en la figura 5.1 y donde se ve que es turno del jugador. Para el ejemplo se está utilizando el nivel *Easy*, que es el más sencillo de los tres. Aquí ya empieza el juego.

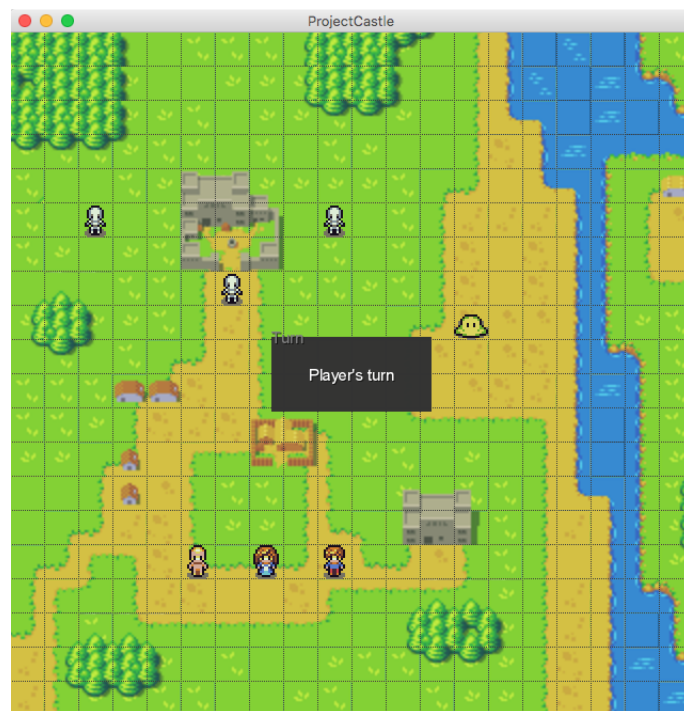


Fig. 5.1. Pantalla de inicio del primer mapa en nivel *Easy*

Para empezar a jugar, se tiene que pulsar en cualquier parte para quitar la ventana del

turno. Luego, si el jugador pulsa sobre un héroe aparecerá en el mapa el menú de acciones así como la pantalla de información con todos los atributos del héroe seleccionado, como se puede apreciar en la figura 5.2.

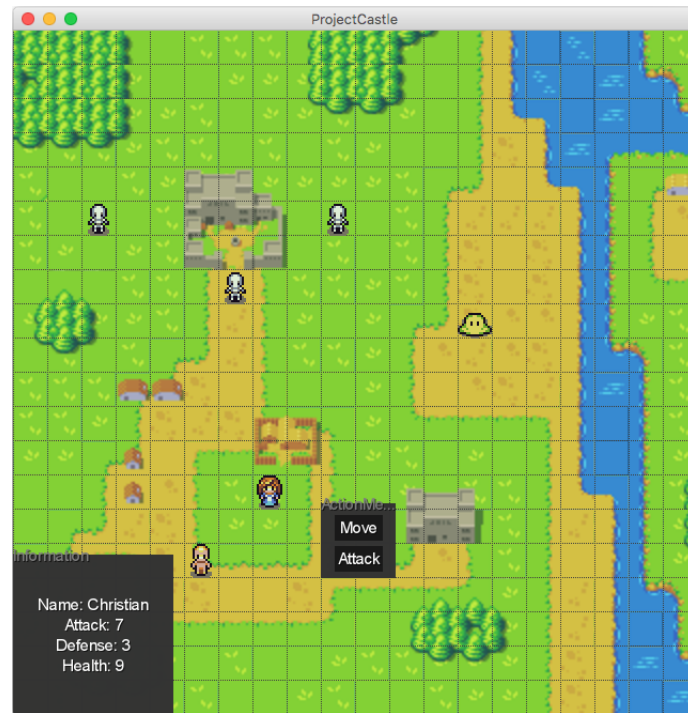


Fig. 5.2. Imagen luego de pulsar sobre un héroe

Si lo que se quiere es mover al personaje, se pulsa sobre el botón *Move* y a continuación se pintan de otro color las casillas donde el héroe se puede mover (figura 5.3). Para proceder a mover al héroe, se pulsa sobre la casilla a la que se quiere que el mismo sea desplazado. En el caso de que se quiera mantener al héroe en esa posición, es necesario pulsar fuera de la zona de movimiento del mismo.

En los juegos de estrategia es importante conocer también la información de los enemigos. Para ello, solo es necesario pulsar sobre uno de ellos y aparecerá en el mismo sitio que la de los héroes. También se podrá ver el radio de movimiento del mismo. En la figura 5.4 se muestra un ejemplo de la selección de un enemigo.

Para acabar el turno, todos los ataques posibles se tienen que haber realizado. Para atacar se tiene que colocar al héroe en una casilla adyacente al enemigo, pulsar sobre el héroe, aparecerá de nuevo el menú de acciones, tocar el botón *Attack* y luego sobre el enemigo que se quiera atacar. El ataque tiene un porcentaje de éxito del 85 %; en caso de que haya funcionado aparecerá sobre el enemigo, en rojo, el daño que se le ha infringido (figura 5.5) y, en caso de que haya fallado, aparecerá un guion negro sobre él.

Este es un juego de rol, y como tal, tiene que existir ese principio básico del que se habló en el apartado 3.1: “un videojuego es del género RPG si el jugador tiene que desarrollar a uno o varios personajes persistentes a través de decisiones tomadas por él, y que estas tengan efecto directo sobre el juego”. En el caso de *Overworld Tactics* se



Fig. 5.3. Movimiento del héroe seleccionado

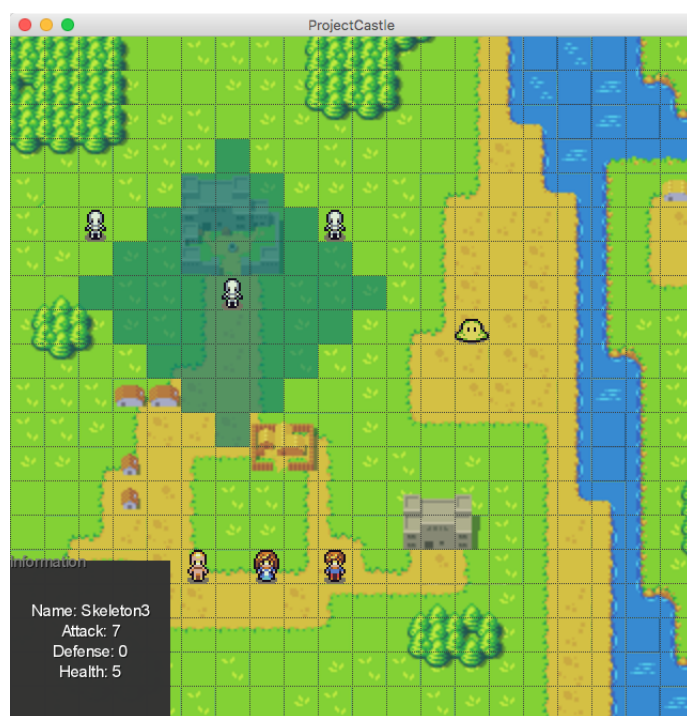


Fig. 5.4. Información del enemigo



Fig. 5.5. Demostración de ataque. Ha quitado 3 de daño al enemigo

puede desarrollar a los personajes mediante la subida de los atributos de ataque y defensa siguiendo unas reglas explicadas en el apartado homónimo del capítulo anterior.

Los enemigos también se mueven en sus turnos. El movimiento y ataque de los mismos viene dado por un código de reglas sencillas que es mejor explicar a través de un diagrama de flujo, como el de la figura 5.6. En resumen, los enemigos atacan a los héroes que puedan matar en un solo golpe, o bien, al que tenga la defensa más baja. Buscan primero dentro de su rango de ataque y si no hay ninguno, se acercan al que cumpla con las condiciones anteriores.

5.2. Descripción de las clases

Overworld Tactics está desarrollado en Java y es compatible con Windows, MacOS y Linux que tengan instalada la versión 1.6 de Java o superior.

El juego está compuesto por 19 clases y 1903 LOC (siglas de líneas de código en inglés), contadas a través de *GitHub*.

A continuación se explicará el funcionamiento de cada una de las clases que forman parte del videojuego, así como los métodos y atributos. El signo “+” antes del nombre de un atributo o método representa que es público, el signo “-” que es privado, el signo “#” que es protegido (solo se puede acceder desde la misma clase y sus herederas) y si no tiene signo es que se puede acceder desde dentro del paquete. En la figura 5.7 se puede visualizar el diagrama de clases antes de mirar en detalle cada una de ellas. Cabe destacar que la clase *Constants* estaría conectada con prácticamente todas, ya que incluye todas

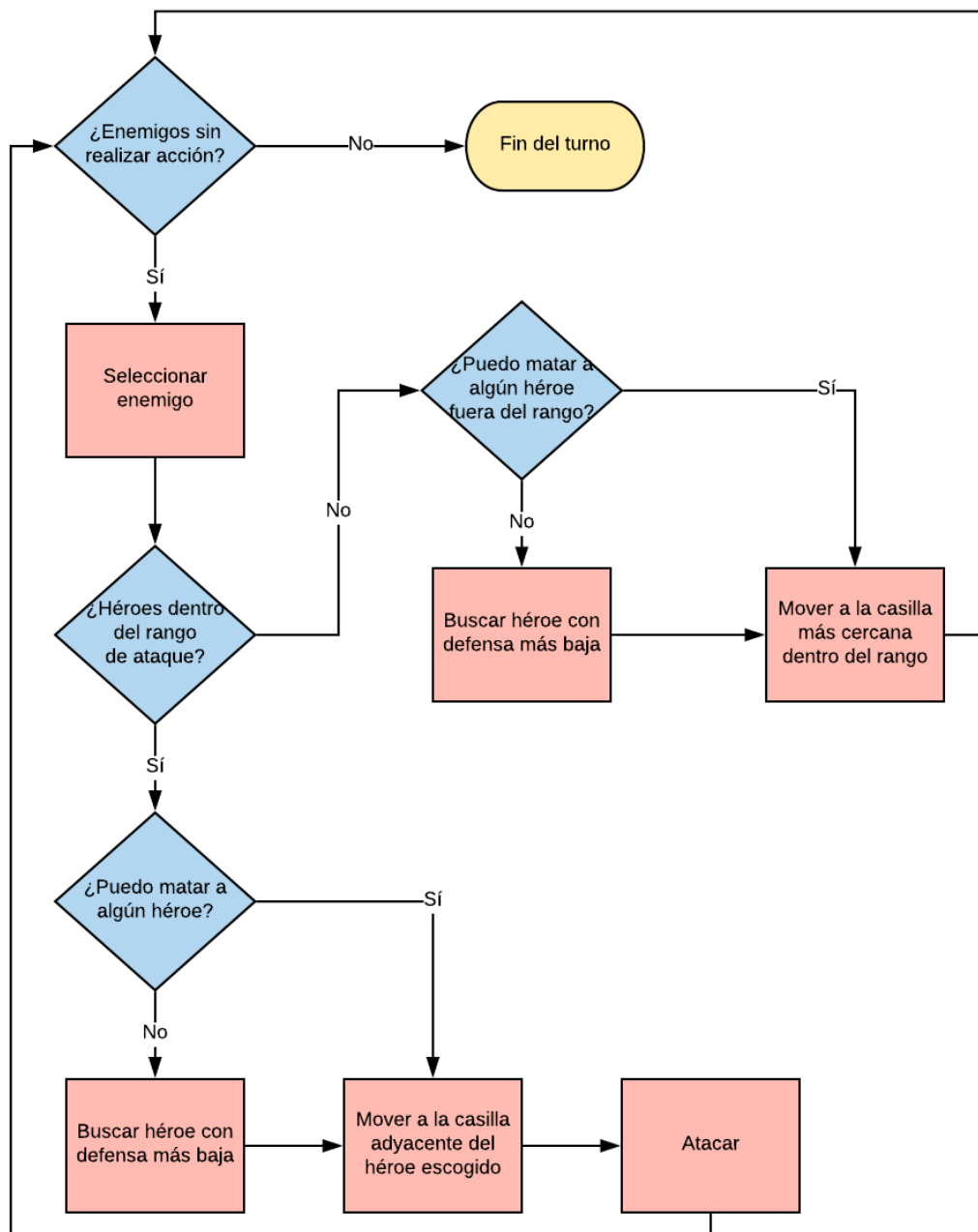


Fig. 5.6. Diagrama de flujo de la toma de decisiones de los enemigos

las constantes utilizadas en el programa, por lo que por decisión de diseño se mantuvo separada para mejor lectura del diagrama.

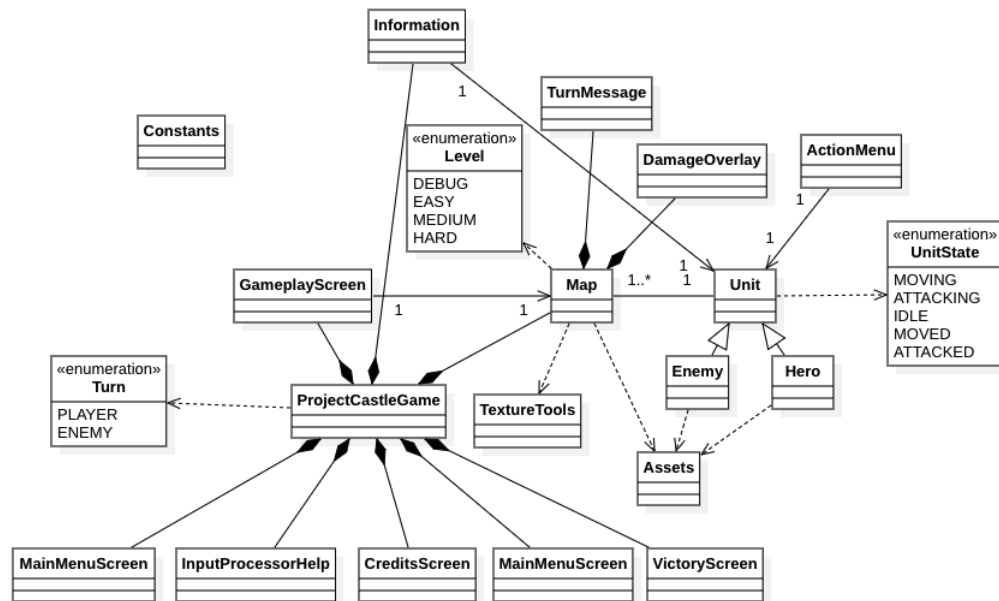


Fig. 5.7. Diagrama de clases de *Overworld Tactics*

Todos los métodos *getters* y *setters*, así como los constructores, serán ignorados para facilitar la lectura, a no ser que el método realice algo más que simplemente devolver o cambiar una variable.

5.2.1. ProjectCastleGame

- Descripción: clase principal que gestiona el bucle del juego y guarda variables importantes.
- Atributos:
 - -fpsLogger: imprime en la consola los fotogramas por segundo a los que se ejecuta el juego.
 - +batch: dibuja todos los elementos visibles del juego.
 - +manager: gestiona las texturas y elementos de la interfaz de usuario.
 - +actionMenu: es el menú de acciones de los héroes.
 - +turnMessage: ventana de cambio de turno.
 - +information: ventana con la información de las unidades.
 - +skin: elementos de la interfaz de usuario.
 - +activeTurn: turno activo en el momento del juego.
 - +timer: gestiona los tiempos en el juego.

- -mapCont: contador para saber el mapa actual.
- Métodos:
 - +create(): inicializa el juego.
 - +render(): dibuja el juego.
 - +dispose(): destruye los recursos.

5.2.2. Map

- Descripción: clase en la que se gestiona todo lo que ocurre sobre el mapa.
- Atributos:
 - -tiledMap: el mapa importado desde *Tiled*, el editor de niveles utilizado para el proyecto.
 - +gameOver: booleano que determina si se ha perdido.
 - +victory: booleano que determina si se ha ganado.
 - enemies: lista de enemigos.
 - heroes: lista de héroes.
 - textureTools: herramientas para tratar las imágenes y texturas.
 - +stage: gestiona los actores del juego (unidades y elementos de la interfaz de usuario).
 - +viewport: controla el tamaño de la pantalla.
 - +tiledMapRenderer: se encarga de dibujar el mapa en pantalla.
 - +camera: la cámara del juego.
 - +shapeRenderer: se encarga del dibujo de figuras. Las líneas horizontales y verticales del mapa se realizan con este.
 - +game: instancia de ProjectCastleGame para acceder a sus métodos y atributos.
 - selectedTileLayer: capa del mapa que se colorea para mostrar el movimiento.
 - selectedTileSet: conjunto de texturas usadas para colorear el movimiento.
 - selectedSpriteRegion: textura que colorea el rango de movimiento de las unidades.
 - inputProcessorHelp: instancia de InputProcessorHelp.
 - +enemiesNewPositions: posiciones de los enemigos después de la toma de decisiones.
 - +selectedLevel:

■ Métodos:

- +update(delta): actualiza el movimiento de los actores (unidades y elementos de la interfaz de usuario).
- +render(batch): dibuja el mapa, luego las líneas de las casillas y por último los actores.
- -initializeMapObjects(): recoge las unidades del mapa de *Tiled*, las añade a las listas y al *stage*.
- +dispose(): destruye los recursos inicializados en el constructor.
- +debugMap(game): crea el mapa de pruebas y llama a initializeDebugMap.
- -initializeDebugMap(): inicializa el mapa de pruebas
- +highlightTilesToMove(calledBy): resalta las casillas a las que se puede mover la unidad que llama al método.
- -removeNextTiles(cells, positionsToRemove, originX, originY, x, y): es llamado internamente desde el método anterior. Elimina las casillas a las que no se puede mover la unidad, bien sea porque las ocupa otra unidad o porque es agua.
- -createCellsList(x, y, limit, cells): método recursivo que obtiene las celdas adyacentes a x e y, las va guardando en cells. Luego se llama a si mismo restando uno a limit. Con esto se van creando las casillas a las que se puede mover un personaje.
- +clearHighlightedTiles(calledBy): limpia las casillas para moverse de la unidad que llama al método.
- -changeTurn(): cambia al turno activo.
- +verifyTurnChange(): si todos los héroes que podían atacar lo han hecho, cambia el turno.
- -runAI(): se encarga de ejecutar la toma de decisiones de los enemigos uno por uno y cambiar el turno al final.
- +touchDown(screenX, screenY, pointer, button): llama al método MapTouchDown de la clase InputProcessorHelp.

5.2.3. InputProcessorHelp

- Paquete: gameplay
- Descripción: gestiona todo lo que ocurre al pulsar el ratón en el mapa.
- Atributos:
 - -game: instancia de ProjectCastleGame

- Métodos:
 - MapTouchDown(map, screenX, screenY): método en el que se gestiona todo lo que ocurre en el mapa cada vez que se pulsa sobre él. No gestiona los toques sobre las unidades.

5.2.4. Unit

- Paquete: entities
- Descripción: clase que hereda de la clase Actor de *LibGDX* para poder usarse dentro de un *stage* y facilitar el movimiento. Los enemigos y los héroes heredan de esta clase.
- Atributos:
 - -attack: ataque de la unidad.
 - -defense: defensa de la unidad.
 - -name: nombre de la unidad.
 - -salud: salud de la unidad.
 - -region: región de la textura de la unidad.
 - -state: estado de la unidad. Se explicarán más adelante en los Enums.
 - -map: instancia de Map.
 - -canMovePositions: lista de posiciones a la que la unidad se puede mover.
- Métodos:
 - +draw(batch, parentAlpha): dibuja la unidad.
 - setStatsAfterAttack(attackingUnit, defendingUnit): cambia los *stats* de la unidad atacante y defensora
 - +setState(state): si el estado cambia a ATTACKED verifica si se puede cambiar el turno llamando a verifyTurnChange de la clase Map.
 - isAdjacent(attackedUnit): verifica que la unidad que llama al método sea adyacente a la que ha sido atacada.
 - +isAdjacent(callingUnit, attackedUnit): verifica que la posición que manda la unidad que llama al método sea adyacente a la que va a ser atacada.
 - +canAttack(newPosition, stage): verifica que la unidad que llama al método pueda atacar desde una posible posición futura.
 - canAttack(position): devuelve una lista con los héroes a los que el enemigo que llama al método puede atacar.

5.2.5. Hero

- Paquete: entities
- Descripción: clase que hereda de Unit y a la que pertenecen todos los héroes.
- Métodos:
 - +Hero(*atributos de Unit*): constructor de la clase. Se encarga de inicializar al héroe y gestionar lo que ocurre al pulsar sobre él.

5.2.6. Enemy

- Paquete: entities
- Descripción: clase que hereda de Unit y a la que pertenecen todos los enemigos.
- Atributos:
 - -showingInfo: ¿está mostrando la información el enemigo?
- Métodos:
 - +Enemy(*atributos de Unit*): constructor de la clase. Se encarga de inicializar al enemigo y gestionar lo que ocurre al pulsar sobre él.
 - +runAI(): ejecuta la toma de decisiones del enemigo.
 - -moveToAdjacent(heroToAttack): se mueve al enemigo a la casilla adyacente al héroe a atacar.
 - -canAttackInZone(): determina si puede atacar a algún héroe dentro del rango de movimiento.
 - -selectAttack(heroes): se selecciona al héroe al cual atacar.
 - -attackHero(hero): se ataca al héroe seleccionado y se cambia el estado de la unidad a ATTACKED.
 - -selectMove(): una vez se determina que no se puede atacar dentro del rango, esta función consigue la posición más cercana al héroe escogido para acercarse a él.
 - -findHeroToPursue(): devuelve la posición del héroe a atacar.
 - +setShowingInfo(): limpia o pinta las casillas dependiendo de si se pulsa sobre el héroe o fuera. También ocurre lo mismo con la información.

5.2.7. Assets

- Paquete: util
- Descripción: clase que gestiona todos los recursos utilizados. Es que la que se encarga de cargarlos y almacenarlos para ser usados por el resto de clases.
- Atributos:
 - +mapAssets: objeto que almacena los recursos de los mapas.
 - +unitsAssets: objeto que almacena los recursos de las unidades.
 - +selectedAssets: objeto que almacena los recursos de la selección de las casillas.
 - -assetManager: es el gestor de los recursos.
- Métodos:
 - +init(assetManager): método que se encarga de cargar todos los recursos y almacenarlos en sus respectivos objetos.
 - +error(asset, throwable): si no se puede cargar algún recurso se llama a este método, que imprime en consola el error.
 - +dispose(): destruye todos los recursos al hacer dispose del assetManager.

5.2.8. Constants

- Paquete: util
- Descripción: clase en la que se guardan todas las constantes del juego.

5.2.9. Enums

- Paquete: util
- Descripción: archivo en el que se crean todos los enumerados del juego
- Enumerados:
 - UnitState: estados de las unidades.
 - MOVING: se está moviendo.
 - ATTACKING: está atacando.
 - IDLE: no ha realizado ninguna acción.
 - MOVED: se ha movido, pero no ha atacado.
 - ATTACKED: ha atacado o se ha movido a una posición donde no puede atacar.

- Turn: turno del juego.
 - PLAYER: turno del jugador.
 - ENEMY: turno de los enemigos.
- Level: nivel de dificultad escogido.
 - DEBUG: mapa de pruebas.
 - EASY: nivel fácil.
 - MEDIUM: nivel intermedio.
 - HARD: nivel difícil.

5.2.10. TextureTools

- Paquete: util
- Descripción: clase en la que se tratan las texturas.
- Métodos:
 - +positionConverter(x, y): entra por parámetro la posición en casillas y la devuelve en píxeles.
 - +tileFinder(x, y): devuelve en píxeles la posición de la esquina inferior derecha de la casilla en la que se encuentra la posición recibida por parámetro.

5.2.11. ActionMenu

- Paquete: screens
- Descripción: menú de acciones de los héroes.
- Atributos:
 - -calledBy: unidad que es pulsada y abre el menú.
- Métodos:
 - +ActionMenu(skin): constructor de la clase. Inicializa las variables y gestiona las acciones cuando se pulsa sobre los botones.

5.2.12. CreditsScreen

- Paquete: screens
- Descripción: pantalla de créditos.
- Atributos:

- camera: cámara del juego.
 - viewport: gestiona el tamaño de la pantalla.
 - stage: gestiona todas las etiquetas de la pantalla.
 - returnMain: botón para volver a la primera pantalla.
- Métodos:
 - +CreditsScreen(game): inicializa la pantalla y gestiona el botón.
 - +render(delta): dibuja la pantalla.
 - +dispose(): destruye los elementos de la pantalla.

5.2.13. DamageOverlay

- Paquete: screens
- Descripción: número que sale sobre la unidad cuando se ejecuta un ataque. Solo tiene un constructor.

5.2.14. GameOverScreen

- Paquete: screens
- Descripción: pantalla de fin de partida.
- Atributos:
 - +stage: gestiona los elementos de la pantalla.
- Métodos:
 - +render(): dibuja la pantalla.
 - +dispose(): destruye los elementos de la pantalla.

5.2.15. GameplayScreen

- Paquete: screens
- Descripción: pantalla en la que se desarrolla el juego.
- Atributos:
 - -game: instancia de ProjectCastleGame
 - -batch: gestiona el dibujado de objetos.
 - -map: mapa actual que tiene que estar en pantalla.

- Métodos:
 - -startNewMap(level, mapCont): inicializa el mapa actual dependiendo del nivel y el contador de mapa.
 - +render(delta): llama al método render() del mapa y actualiza los cambios en cada ciclo de juego.
 - +dispose(): destruye todos los recursos.

5.2.16. Information

- Paquete: screens
- Descripción: ventana de información que aparece en la parte inferior al seleccionar a una unidad.
- Atributos:
 - -calledBy: unidad seleccionada.
 - -game: instancia de ProjectCastleGame.
- Métodos:
 - +setVisible(visible): método que gestiona lo que se escribe sobre la ventana y que lo borra cuando la misma desaparece.

5.2.17. MainMenuScreen

- Paquete: screens
- Descripción: pantalla principal, donde se selecciona el nivel de dificultad.
- Atributos:
 - -stage: gestiona los elementos de la pantalla.
- Métodos:
 - +MainMenuScreen(game): inicializa todos los elementos de la pantalla y gestiona cuando se pulsa en los botones.
 - +render(delta): dibuja la pantalla principal.
 - +dispose(): destruye los recursos utilizados.

5.2.18. TurnMessage

- Paquete: screens
- Descripción: ventana de cambio de turno.
- Atributos:
 - -turnTimer: gestiona el tiempo que dura la pantalla.
- Métodos:
 - +setTurn(activeTurn): cambia el turno y muestra la pantalla.
 - +setVisible(visible): cuando se llama con estado falso, borra todos los elementos.

5.2.19. VictoryScreen

- Paquete: screens
- Descripción: pantalla de victoria. Para llegar se tienen que vencer los dos mapas de un nivel.
- Atributos:
 - -stage: gestiona los elementos de la pantalla.
- Métodos:
 - +render(delta): dibuja la pantalla.
 - +dispose(): destruye los recursos de la pantalla de victoria.

5.3. Versiones

A continuación se explicarán las versiones del juego realizadas y los cambios introducidos en cada una de ellas.

5.3.1. Versión 0.1.0

- Primera versión “jugable”. Tiene pantalla de inicio, un mapa, el menú de acciones, la ventana de información y los personajes se mueven y atacan.
- Un solo enemigo.
- Figura 5.8



Fig. 5.8. Versión 0.1.0

5.3.2. Versión 0.1.1

- Arreglado un error en la ventana de información en el que se descolocaba el nombre de la unidad seleccionada.

5.3.3. Versión 0.2.0

- Ya se puede ver el daño que se causa al enemigo y viceversa.
- La ventana de información tiene un tamaño más aceptable.

5.3.4. Versión 0.3.0

- Primera versión con el mapa estilo *overworld*. Ya no existen más problemas con unidades que están encima de árboles o trepando la pared del castillo (como en la figura 5.9).
- Tres niveles de dificultad para el mismo mapa.
- Ahora la división de casillas es visible desde el mapa.

5.3.5. Versión 0.4.0

- Ahora la probabilidad de asestar un golpe es del 70 %. Antes siempre se pegaba.



Fig. 5.9. Enemigos “trepando” por las paredes del castillo

5.3.6. Versión 1.0.0

- Primera versión oficial del juego.
- Segundo mapa añadido para cada dificultad (figura 5.10)
- La probabilidad de golpear es ahora del 85 %.
- En el resto de pantallas, ahora están mejor centradas las etiquetas.

5.3.7. Versión 1.0.1

- Arreglado error que una vez el jugador pasaba al segundo mapa y volvía a iniciar la partida desde los créditos, empezaba de nuevo en el segundo mapa, cuando debía ser desde el primero.

5.4. Pruebas y retroalimentación

Para diseñar las pruebas se utilizaron los consejos encontrados en el apartado de *Effective Playtesting* del libro *Advanced Game Design* escrito por Michael Sellers [42]. El autor comenta, entre otras cosas, que la forma más efectiva de realizar las pruebas de un videojuego es a través de la experiencia, y para ello es necesario pedir a otras personas que prueben el juego.



Fig. 5.10. Segundo mapa del juego, en dificultad fácil

Otro punto muy importante que comenta Sellers es que el juego digital tiene que probarse lo antes posible, es por ello que *Overworld Tactics* empezó a ser testado desde la primera versión jugable, eso quiere decir, la versión 0.1.0 y se siguió probando en cada iteración. A diferencia de otras herramientas software, “probar videojuegos no trata de conseguir bugs en tu código” [42], sino de saber si el juego está siendo divertido y entendible para los jugadores.

Teniendo en cuenta lo anterior, se consiguieron tres personas que ayudarían a probar las versiones del videojuego a medida que se iban desarrollando (excepto las versiones 0.1.1 y 1.0.1 por no tener cambios muy importantes) y en la tabla 5.1 se explica la retroalimentación conseguida, en qué versión, la solución a la misma y la versión en que se arregló cada problema.

Además de lo anterior, se envió la versión final del juego, junto con el manual de usuario (Anexo A) a 13 personas y se les pidió realizar la encuesta (elaborada en *Google Forms* [43]) que se puede encontrar en el Anexo B de este documento.

Al ser una encuesta tan pequeña y sesgada, no se pueden sacar grandes conclusiones de la misma. Pero aún así, arroja ciertas relaciones que se pueden tomar en cuenta, como las que se explicarán a continuación.

Para un videojuego, lo más importante es saber si la gente lo considera divertido. Es por eso que una de las preguntas de la encuesta es precisamente esa, con opciones que van del 1 (muy aburrido) al 5 (muy divertido). En la figura 5.11 se aprecia que el promedio por cada tipo de jugador está por encima de tres, llegando a ser cuatro en aquellas personas que juegan más de diez horas a la semana, sin embargo no es mala nota un cuatro entre

Versión	Retroalimentación	Solución	Versión solución
0.1.0	Muy fácil y simple	Añadir más enemigos	0.2.0
0.1.0	La información se muestra de forma extraña	Arreglar ventana de información	0.1.1
0.1.0	No se ve el daño causado	Añadir el daño causado al hacer un ataque	0.2.0
0.2.0	Las unidades caminan sobre paredes y árboles	Cambiar el estilo de mapa a uno del tipo overworld	0.3.0
0.2.0	El juego es muy simple	Añadidas tres dificultades	0.3.0
0.3.0	Es muy determinista para el género al que pertenece	Probabilidad del 70 % de atacar con éxito	0.4.0
0.3.0	Un solo mapa con tres dificultades es muy poco	Creado un segundo mapa al acabar el primero en las tres dificultades	1.0.0
0.4.0	Muchos fallos al atacar en una partida	Se subió la probabilidad de atacar al 85 %	1.0.0
1.0.0	Al acabar la partida y volver a empezar, se inicia desde el segundo mapa	Error arreglado	1.0.1

TABLA 5.1. PROBLEMAS ENCONTRADOS Y SOLUCIONES

aquellos que van de 1 a 9 horas jugadas semanales. Por tanto, se podría decir que un público casual vería divertido el juego. Solo una persona contestó que no juega a videojuegos, por lo que no es posible generalizar de su respuesta.

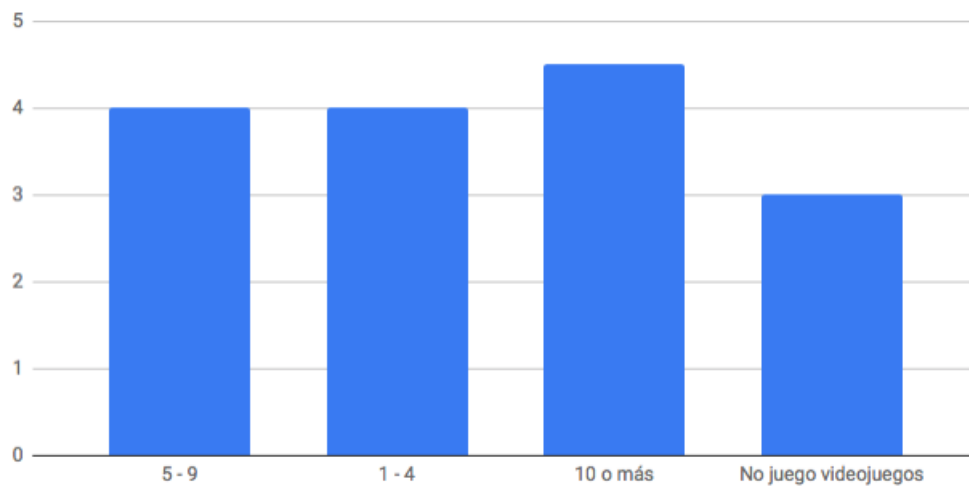


Fig. 5.11. Número de horas jugadas a la semana versus qué tan divertido considera el juego

Otros puntos interesantes a tomar en cuenta serán los promedios obtenidos de la valoración de ciertos apartados de *Overworld Tactics*. En el caso de la interfaz de usuario, en una puntuación del 1 al 5, se obtuvo un promedio de 3,77; mientras que en el estilo artístico un 4,46. Lo cual no sorprende ya que la interfaz puede mejorar mucho más, mientras que las texturas escogidas para el videojuego realmente están muy bien hechas, y además los mapas creados a partir de ellas parecen ser del agrado de las personas.

En los juegos de estrategia, saber si la dificultad está bien balanceada es importante. Por ello existían tres preguntas en las cuales se pedía la opinión con respecto a si correspondía el nombre de la dificultad con la misma en una escala del 1 (muy fácil) al 5 (muy difícil). Las respuestas en este apartado fueron muy variadas y se encuentran en la figura 5.12

Las dificultades estarían bien balanceadas si la gran mayoría de las barras se acercaran al número tres. Sin embargo, en el caso de la dificultad fácil hay muchas barras en el uno y en el dos, no quiere decir que sea malo, pero podría aumentarse un poco la dificultad. Algo parecido ocurre con las barras amarillas, que corresponden a difícil, hay dos que llegan al cinco y otras dos que se quedan en el cuatro. En el caso de la dificultad media, parece que dentro de este grupo de personas, se corresponde perfectamente.

Otros datos interesantes extraídos de la encuesta:

- Solo una persona no pudo completar el juego en ninguna dificultad.
- El rango de edades de la muestra va desde los 18 hasta los 35 años.
- A la única persona que no juega videojuegos le pareció que las tres dificultades eran difíciles (4, 4, 5).

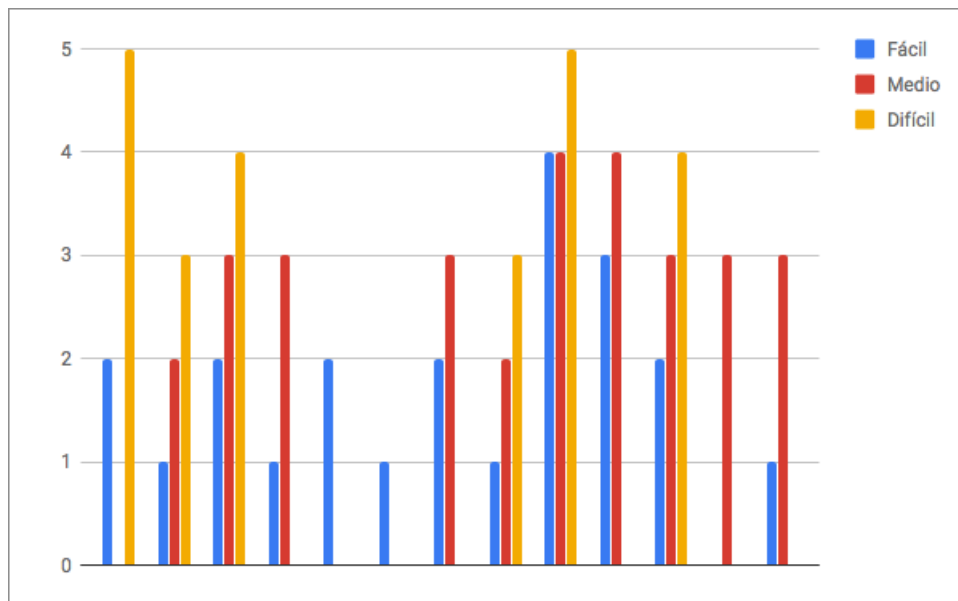


Fig. 5.12. Dificultad de *Overworld Tactics*

- No se sacaron conclusiones respecto al género, puesto que solo una mujer contestó a la encuesta.
- La nota media (del 1 al 10) del juego resultó ser 7,23.

Adicionalmente a lo comentado por Sellers, también se ejecutaron pruebas sobre todas las mecánicas mencionadas en el apartado de diseño a medida que se iban programando y hasta que no salieran exitosas, no se seguía con la siguiente. En la tabla 5.2 se pueden ver todas estas.

Descripción prueba	Resultado esperado	Resultado obtenido
Pantalla de selección	Al tocar un botón de dificultad envía a la pantalla correcta	Correcto
Menú de acciones	Al tocar sobre un héroe aparece el menú de acciones	Correcto
Rango movimiento	Al tocar sobre el botón Move aparece el rango de movimiento del héroe	Correcto
Agua	Las casillas con agua no aparecen en el rango de movimiento	Correcto
Mover	Al tocar dentro del rango de movimiento se mueve el héroe	Correcto
No mover	Al tocar fuera del rango el héroe no se moverá más durante el turno	Correcto
Mover una vez	No podrán moverse más de una vez los héroes en un turno	Correcto
Atacar desde lejos	Si se toca Attack sin estar adyacente a un enemigo, no pasa nada	Correcto
Atacar	Al tocar Attack y luego a un enemigo adyacente al héroe se realiza un ataque	Correcto
Fallo	Los ataques pueden fallar	Correcto
Atacar antes que mover	Si se ataca sin mover, luego no se podrá mover	Correcto
Tocar héroe inhabilitado	El menú de acciones no tiene que aparecer	Correcto
Daño causado	Sobre la unidad atacada aparecerá en rojo la salud quitada o un guion negro en caso de fallo	Correcto
Fin de turno	Si todos los héroes realizan sus acciones cambia el turno	Correcto
Ventana de información	Aparece siempre que se toque sobre cualquier unidad con los datos de esta	Correcto
Rango enemigos	El rango de movimiento del enemigo aparecerá cuando se toque sobre él	Correcto
Subida ataque	El ataque de una unidad aumenta según las reglas al atacar	Correcto
Subida defensa	La defensa de una unidad aumenta según las reglas al atacar	Correcto
Reducción salud	La salud de la unidad atacada se reduce la diferencia entre el ataque de la atacante y la defensa de esta	Correcto
Muerte	Si la salud de una unidad llega a cero, desaparece	Correcto
Comportamiento enemigos	Los enemigos se comportan de acuerdo a las reglas mencionadas en el apartado de diseño	Correcto
Paso a segundo mapa	Al acabar la primera pantalla, se pasará a la segunda automáticamente	Correcto
Fin de la partida	Si los héroes mueren, sale la pantalla de Game Over	Correcto
Fotogramas por segundo	El número de fotogramas por segundo es de 60 constantes	Correcto

TABLA 5.2. PRUEBAS EJECUTADAS

6. ENTORNO SOCIO-ECONÓMICO

En este apartado se estudiará la planificación y los costes del proyecto teniendo en cuenta las horas invertidas en él, así como los recursos utilizados. También se hablará del impacto socio-económico del mismo y por último se analizará el marco regulador.

6.1. Planificación

Para la planificación de este proyecto, se creó un diagrama de Gantt utilizando la herramienta [36] y que se puede ver en la figura 6.1. Debido a que al inicio de este proyecto no se había trabajado nunca con *LibGDX*, se estimó una formación larga y suficientes horas en cada objetivo hasta lograr las 300 correspondientes a los doce créditos europeos.

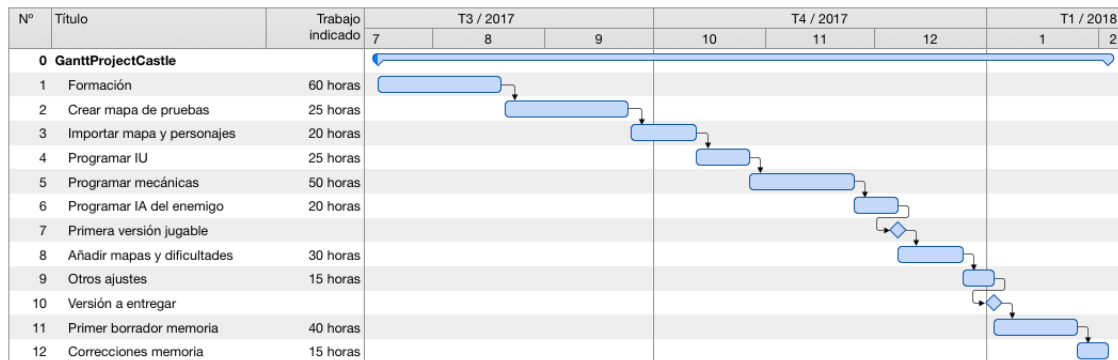


Fig. 6.1. Gantt inicial del proyecto

El significado de cada uno de los puntos del diagrama, se encuentra explicado en la tabla 6.1.

En principio se pretendía trabajar dos horas al día de lunes a sábado y así se programó en la herramienta *Merlin Project Express*, con lo que al colocar las horas estimadas, esta creaba el diagrama según el tiempo que se le podía dedicar. También permitía añadir excepciones, por lo que las fechas del 26 al 31 de agosto y del 3 al 9 de octubre no son contadas en el diagrama por vacaciones.

Lamentablemente el diagrama no se pudo seguir estrictamente. Las dos razones principales de esto son que algunos días no se le podían dedicar las dos horas reglamentarias al proyecto, como que la estimación de horas inicial no resultó adecuada en algunos casos. El trabajo real es el que se puede ver en la figura 6.2

Viendo ambos diagramas, se podrían sacar algunas conclusiones muy interesantes. Lo primero que se puede observar es que se pretendía acabar el Trabajo de Fin de Grado en febrero de 2018 para acceder a la convocatoria de marzo, sin embargo no fue posible por las razones explicadas en el párrafo anterior (no se pudo trabajar dos horas diarias y la estimación de horas no era adecuada).

Objetivo	Descripción
Formación	Tiempo dedicado a aprender a utilizar LibGDX y Tiled.
Crear mapa de pruebas	Creación de un primer mapa en el cual programar las mecánicas del juego usando Tiled.
Importar mapa y personajes	Importación del mapa creado anteriormente a LibGDX, así como la adición de los personajes.
Programar IU	Desarrollo de la interfaz de usuario del juego. Esto incluye todas las pantallas del juego, así como el menú de acciones y la ventana de información.
Programar mecánicas	Desarrollo de las mecánicas principales del videojuego. Mover y atacar con los personajes, así como la comprobación de su correcto funcionamiento.
Programar IA del enemigo	Desarrollo de la toma de decisiones de los enemigos y la gestión de los turnos.
Añadir mapas y dificultades	Creación de dos mapas y tres niveles de dificultad.
Otros ajustes	Mejoras en general del juego. Balance de las unidades, cambios en los sprites.
Primer borrador memoria	Creación del primer borrador a ser revisado por el tutor del Trabajo de Fin de Grado.
Correcciones memoria	Correcciones necesarias a la memoria según lo acordado con el tutor.

TABLA 6.1. OBJETIVOS Y DESCRIPCIÓN DEL DIAGRAMA DE GANTT

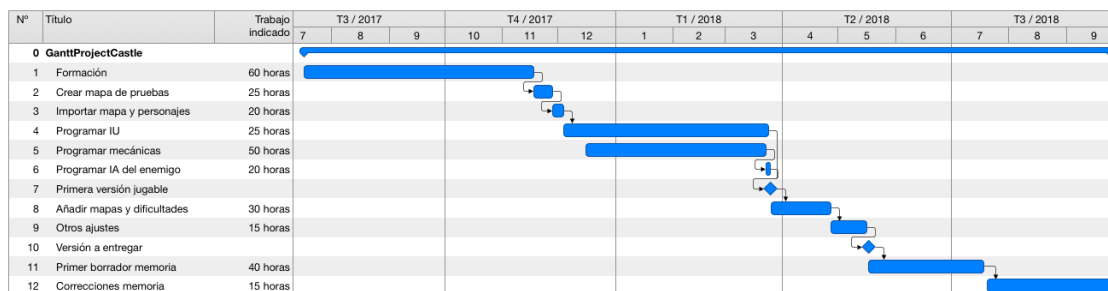


Fig. 6.2. Gantt real del proyecto

También se ve que la formación en la herramienta tomó mucho más tiempo del deseado acabando en noviembre, en lugar de agosto como en principio se quería, lo cual ya generó tres meses de retraso. Otro punto a destacar es que algunas tareas fueron más sencillas de lo que se creía, como por ejemplo la creación del mapa de pruebas, su importación al código y la de los personajes, así como la programación de la toma de decisiones de los enemigos.

Por último, también es notable el cambio en la ruta crítica. Mientras que en el primer diagrama es todo secuencial, a la hora de desarrollar se hizo necesario paralelizar la programación de la interfaz de usuario y de las mecánicas de juego, ya que a medida que se iban avanzando por una parte, se requería también que la otra tuviera desarrollo. Un ejemplo es que para mover y atacar se necesitaba que los botones del menú de acciones funcionaran.

6.2. Costes

Para evaluar los costes, el proyecto se ha dividido en tres grandes etapas: formación, desarrollo y documentación. Se han utilizado las aplicaciones *Timing* [34] y *Hours* [35] para registrar el número de horas.

El coste por hora se tomó de la encuesta que *Stack Overflow* realiza todos los años a los desarrolladores [44]. Se calculó el salario por hora teniendo en cuenta el resultado de la encuesta para los desarrolladores de videojuegos y se concluyó que son aproximadamente 17,86€.

Teniendo todos estos datos, se construyó la tabla 6.2, en la que se reflejan los costes totales de los recursos humanos.

Etapas	Horas acumuladas	Coste
Formación	44	785,84 €
Desarrollo	181	3.232,66 €
Documentación	51	910,86 €
Totales	276	4.929,36 €

TABLA 6.2. COSTES TOTALES DE LOS RECURSOS HUMANOS

Para los costes materiales se tomó en cuenta el valor de compra de los productos y servicios, lo que duraría su amortización y el número de meses en los que se le dio uso con el proyecto. El resultado se puede apreciar en la tabla 6.3. No se tomaron en cuenta todas las licencias de *Software* gratuitas.

Sumando ambos resultados, el subtotal del proyecto es de 5.160,83 €. Sin embargo, a esto se le suma un 15 % de costes indirectos y un 33 % de Seguridad Social (cifra conseguida en [45]); lo que hace que el coste total de *Overworld Tactics* ascienda a unos 7.638,03 €. Esta última cuenta puede verse desglosada en la tabla 6.4.

Producto	Precio	Meses de Amortización	Coste al mes	Meses usado en el proyecto	Coste en el proyecto
MacBook Pro 13” mediados del 2012	980,00 €	72	13,61 €	13	176,94 €
Silla reclinable 180 grados	85,00 €	36	2,36 €	7	16,53 €
Monitor Dell U2417H	240,00 €	60	4,00 €	6	24,00 €
Suscripción Overleaf v2	14,00 €	2	7,00 €	2	14,00 €
				Total	231,47 €

TABLA 6.3. COSTES TOTALES DE LOS RECURSOS MATERIALES

Recursos	Coste
Humanos	4.929,36 €
Materiales	231,47 €
Subtotal:	5.160,83 €
Costes indirectos (15 %)	774,13 €
Seguridad Social (33 %)	1.703,07 €
Total:	7.638,03 €

TABLA 6.4. COSTE TOTAL DEL PROYECTO

6.3. Impacto socio-económico

La finalidad de este proyecto desde el primer momento ha sido aprender a desarrollar un videojuego. Por lo tanto, no se pretende obtener beneficios así que si la memoria es publicada, será con licencia *Creative Commons*. En el caso del videojuego, será publicado también de forma gratuita para que cualquier persona pueda disfrutar de él.

El código también será liberado en *GitHub* por si existen personas interesadas en continuar con el proyecto.

6.4. Marco Regulator

En cuanto al ámbito legal, no existen leyes en España que regulen el desarrollo y publicación de videojuegos, excepto las referidas a la piratería y a la protección de datos.

El juego será publicado de forma gratuita y con licencia de CC BY 4.0. Lo que significa que se podrá modificar, distribuir de forma comercial o gratuita siempre y cuando se atribuya al autor original. Esta memoria, en cambio, tiene licencia que no permite distribución comercial ni derivada y siempre tiene que atribuirse el trabajo al autor.

En cuanto a la protección de datos, *Overworld Tactics* no recogerá datos de ningún jugador, por lo que el Reglamento General de Protección de Datos no es aplicable.

Con respecto a las herramientas utilizadas, tanto *LibGDX*, como *Tiled* permiten el uso gratuito para obras comerciales. Toda el arte que se ha usado en el videojuego está también bajo licencia *Creative Commons* y permiten la distribución de la misma mientras haya atribución. En los créditos se encuentran los autores de cada imagen y también en las referencias de este trabajo.

7. CONCLUSIONES

La importancia de la industria de los videojuegos en la actualidad es innegable. Económicamente se puede demostrar esto viendo que en el año 2017 generó más ingresos en España que las industrias del cine y la música [1]. Pero no solo afecta a la economía, también el ocio es muy importante en la vida de las personas casi tanto como trabajar o descansar. Es por eso que se tomó la decisión de realizar este proyecto, cuya finalidad no es más que la de aprender a desarrollar un juego digital dirigido exclusivamente al ocio y disfrute de las personas.

Overworld Tactics es un juego de rol estratégico compuesto por dos mapas y tres niveles de dificultad. Es un videojuego sencillo que se puede completar en un periodo de tiempo que va desde 5 a 40 minutos dependiendo del nivel escogido y la cantidad de tiempo dedicado a pensar una estrategia para vencer. Pero a pesar de su corta duración, puede ser rejugable probando cada vez nuevas y mejores tácticas.

También es importante en estas conclusiones hablar de los límites y problemas encontrados a lo largo del proyecto. El principal obstáculo fue la falta de tiempo para dedicar diariamente al juego y la fecha de entrega cada vez acercándose más, razones que hicieron que los objetivos planeados al principio no se pudieran cumplir hasta reducirse a lo actual. También la ausencia de experiencia con la herramienta *LibGDX* hizo que la formación tardara más de lo previsto, así como cada nueva tarea. Por ejemplo: desarrollar un nuevo objetivo podía ocupar una o dos horas de tiempo, en contraposición de la semana, o incluso dos, de investigación para aprender a programarlo. Pero la experiencia hizo que este último problema tuviera un menor peso, es por ello que la diferencia más grande entre la versión 0.3.0 y 0.1.0 no es el cambio de mapa, sino que la estructura del juego se modificó internamente por completo, lo cual facilitó mucho más la adición de niveles y dificultades, así como el número de unidades por mapa.

Viendo hacia el futuro, el juego todavía puede crecer más. Se le puede añadir una historia explícita, más niveles, nuevas formas de interacción, nuevos héroes, distintos tipos de armas, entre otras mejoras. Es por ello que se mantendrá de forma pública en *GitHub* para el que quiera seguir desarrollando sobre él.

BIBLIOGRAFÍA

- [1] Asociación Española de Videojuegos, *Anuario 2017. Anuario de la Industria del Videojuego*. 2018. [En línea]. Disponible en: http://www.aevi.org.es/web/wp-content/uploads/2018/07/AEVI_Anuario2017.pdf.
- [2] B. Loguidice y M. Barton, *Vintage Game Consoles*. Burlington: Focal Press, 2014.
- [3] M. V. Wilkes y W. Renwick, «The EDSAC (Electronic Delay Storage Automatic Calculator)», *Mathematical Tables and Other Aids to Computation*, vol. 4, n.º 30, p. 61, 1950. doi: 10.2307/2002769. [En línea]. Disponible en: <http://www.jstor.org/stable/2002769?origin=crossref>.
- [4] *Oxo*, 2007. [En línea]. Disponible en: <https://upload.wikimedia.org/wikipedia/commons/9/9b/Oxo.jpg>.
- [5] J. Ito, *Spacewar running on PDP-1*, 2007. [En línea]. Disponible en: <https://commons.wikimedia.org/wiki/File:Spacewar!-PDP-1-20070512.jpg>.
- [6] MBERTRAND, *La evolución de las videoconsolas en 8 generaciones y más de 40 dispositivos (infografía)*, 2016. [En línea]. Disponible en: <https://www.nobbot.com/redes/evolucion-de-las-videoconsolas-infografia/>.
- [7] I. Cano, «Nintendo NES, la consola que salvó a los videojuegos de la desaparición», *ABC*, 2013. [En línea]. Disponible en: <http://www.abc.es/tecnologia/videojuegos/20130715/abci-nintendo-aniversario-201307110910.html>.
- [8] M. Cabezas, *La Nintendo 64 cumple 22 años: estos son algunos de sus mejores juegos*, 2018. [En línea]. Disponible en: <http://www.t13.cl/noticia/tendencias/la-nintendo-64-cumple-22-anos-estos-son-algunos-sus-mejores-juegos>.
- [9] J. Cano, *La historia de PlayStation 2*, 2016. [En línea]. Disponible en: <https://vandal.elespanol.com/reportaje/la-historia-de-playstation-2/4>.
- [10] VGChartz Ltd., *Platform Totals*, 2018. [En línea]. Disponible en: http://www.vgchartz.com/analysis/platform_totals/Hardware/Global/.
- [11] A. Williams, «Microsoft's Entry», en *History of Digital Games*, Boca Raton: Focal Press, 2017, cap. 9.
- [12] S. Andrews, *Is this the rebirth of motion controls?*, 2018. [En línea]. Disponible en: <https://www.trustedreviews.com/news/rebirth-motion-controls-3383440>.

- [13] S. Cejas, *El sector de los videojuegos en España ha facturado 1.359 millones en 2017, un 16,9 % más con respecto al año pasado*, 2018. [En línea]. Disponible en: <https://www.vidaextra.com/industria/sector-videojuegos-espana-ha-facturado-1-359-millones-2017-17-respecto-al-ano-pasado>.
- [14] A. Williams, «The Scratchware Manifesto and Dimensions of “Indie”», en *History of Digital Games*, Boca Raton: Focal Press, 2017, cap. 10.
- [15] SteamSpy, 2018. [En línea]. Disponible en: <http://steamspy.com/year/>.
- [16] elotrolado.net, *Géneros de los videojuegos*. [En línea]. Disponible en: https://www.elotrolado.net/wiki/G%C3%A9neros_de_los_videojuegos.
- [17] C. Stern, *What makes an RPG an RPG: a universal definition*, 2011. [En línea]. Disponible en: <http://sinisterdesign.net/what-makes-an-rpg-an-rpg-a-universal-definition/>.
- [18] Square Enix Inc., *Final Fantasy Tactics*, 1997.
- [19] Intelligent Systems, *Fire Emblem*, 2004.
- [20] F. M. López, *Desarrollo de un videojuego RPG con el framework LibGDX*, Trabajo fin de máster, 2014. [En línea]. Disponible en: <http://www.plg.inf.uc3m.es/~dborrajo/pfc/TFM-Francisco-Lopez.pdf>.
- [21] H. A. Huerta, *Diseño de un juego multijugador en Android*, Trabajo fin de grado, 2012. [En línea]. Disponible en: <https://core.ac.uk/download/pdf/30046893.pdf>.
- [22] P. Heinrich y J. Silver, *2D Game Development with LibGDX*. [En línea]. Disponible en: <https://eu.udacity.com/course/2d-game-development-with-libgdx--ud405>.
- [23] ———, *How to Make a Platformer Using LibGDX*. [En línea]. Disponible en: <https://eu.udacity.com/course/how-to-make-a-platformer-using-libgdx--ud406>.
- [24] JetBrains s.r.o., *IntelliJ IDEA*, 2018. [En línea]. Disponible en: <https://www.jetbrains.com/>.
- [25] Badlogic Games, *LibGDX*, 2013. [En línea]. Disponible en: <https://libgdx.badlogicgames.com>.
- [26] T. Lindeijer, *Tiled*, 2016. [En línea]. Disponible en: <https://www.mapeditor.org>.
- [27] Apple Inc., *Safari*, Cupertino, 2018. [En línea]. Disponible en: apple.es.
- [28] Safari Books Online, *Safari Books Online*, 2018. [En línea]. Disponible en: <https://www.safaribooksonline.com>.
- [29] Stack Exchange Inc., *Stack Overflow*, 2018. [En línea]. Disponible en: <https://stackoverflow.com>.

- [30] Lucid Software Inc., *Lucidchart*, 2018. [En línea]. Disponible en: <https://www.lucidchart.com/>.
- [31] Writelatex Ltd., *Overleaf v2*, London, 2018. [En línea]. Disponible en: <https://v2.overleaf.com/>.
- [32] GitHub, *GitHub*, 2018. [En línea]. Disponible en: github.com.
- [33] Mendeley Ltd., *Mendeley*, New York, 2016. [En línea]. Disponible en: <https://www.mendeley.com>.
- [34] Timing, *Timing*, Alemania, 2018. [En línea]. Disponible en: <https://timingapp.com>.
- [35] Hours LCC, *Hours*, 2018. [En línea]. Disponible en: <https://www.hourstimetracking.com>.
- [36] ProjectWizards, *Merlin Project Express*, 2018. [En línea]. Disponible en: <https://www.projectwizards.net/es>.
- [37] L. González, *How to Write a Game Design Document*, 2016. [En línea]. Disponible en: https://www.gamasutra.com/blogs/LeandroGonzalez/20160726/277928/How_to_Write_a_Game_Design_Document.php.
- [38] C. Macklin y J. Sharp, «The Game Design Document», en *Games, Design and Play: A Detailed Approach to Iterative Game Design*, Addison-Wesley Professional, 2016, cap. 7.
- [39] L. Zimmerman, *Tiny 16: Basic*, 2013. [En línea]. Disponible en: <https://opengameart.org/content/tiny-16-basic>.
- [40] Buch y J. Kern, *Overworld Tiles*, 2014. [En línea]. Disponible en: <https://opengameart.org/content/overworld-tiles-0>.
- [41] Czyzby, *Flat skin*, 2016. [En línea]. Disponible en: <https://github.com/czyzby/gdx-lml>.
- [42] M. Sellers, «Effective Playtesting», en *Advanced Game Design*, Addison-Wesley Professional, 2017, cap. 12.
- [43] Google, *Google Forms*, Estados Unidos, 2018. [En línea]. Disponible en: <https://forms.google.com>.
- [44] Stack Exchange Inc., *Stackoverflow Developer Survey 2018*, 2018. [En línea]. Disponible en: <https://insights.stackoverflow.com/survey/2018#salary>.
- [45] M. Empresa, *Las cotizaciones a la Seguridad Social: conceptos, bases y porcentajes*, 2018. [En línea]. Disponible en: <https://www.sage.com/es-es/blog/las-cotizaciones-a-la-seguridad-social-conceptos-bases-y-porcentajes/>.
- [46] Ranker, *The Best Tactical Role-Playing Games of All Time*, 2018. [En línea]. Disponible en: <https://www.ranker.com/list/all-tactical-role-playing-games-list/reference>.

ANEXO A. MANUAL DE USUARIO

Selección de nivel

El juego tiene tres niveles de dificultad: fácil, medio y difícil. Las estadísticas de los héroes no cambian pero sí lo hacen las de los enemigos. Además varía el número de enemigos en el nivel difícil. Si es tu primera vez jugando un RPG estratégico, recomendamos encarecidamente empezar con el nivel fácil.

Cómo jugar

El principal objetivo de *Overworld Tactics* es derrotar a todos los enemigos en ambas pantallas.

En el turno del jugador, tienes dos opciones que puedes realizar con cada uno de tus personajes: mover y atacar. Cada vez que tocas a un héroe, aparece un menú con estos dos botones. Para atacar a un enemigo, tienes que estar en una casilla adyacente, por lo que el procedimiento normal es mover al personaje cerca del enemigo y atacar cuando está en la casilla adyacente.

Cuando el ataque es exitoso, un número rojo aparece sobre la unidad atacada, ese es el daño ocasionado. Si aparece un guion negro, el ataque ha fallado. La probabilidad de asestar el golpe es del 85 %.

Cada unidad puede moverse solo una vez en cada turno, sin embargo, si atacas antes de moverte, ya no lo puedes hacer. Así que piensa bien tu estrategia. Además, cada vez que abras el menú, podrás ver la información de tu unidad. Para ver la información de tus enemigos, puedes tocarlos y aparecerá.

Si en un turno no quieres mover ni atacar con ninguna unidad, puedes dejarla en el mismo sitio tocando sobre el botón mover y luego pulsando fuera del área del movimiento.

En su turno, los enemigos se acercarán hacia tus héroes y atacarán cuando estén en su rango. Las reglas de movimiento y ataque de estos son las mismas que las tuyas y conocen la misma información de tus unidades que tú las de ellos. Se mueven pensando en tus estadísticas.

Tus atributos (y los de los enemigos) crecen a medida que atacas o defiendes, siéntete libre de mirar los cambios cada vez que se abre la ventana de información.

ANEXO B. ENCUESTA PRUEBAS

La encuesta realizada a las personas para el apartado de pruebas se encuentra en las siguientes tres páginas.

6. ¿En cuáles niveles de dificultad jugó? *

Selecciona todos los que correspondan.

- ☐ Fácil
- ☐ Medio
- ☐ Difícil

7. Valore la difficoltà fácil

Marca solo un óvalo.

	1	2	3	4	5	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

8. Valore la difficoltà media

Marca solo un óvalo.

	1	2	3	4	5	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

9. Valore la difficoltà difícil

Marca solo un óvalo.

	1	2	3	4	5	
Muy fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy difícil

10. ¿Logró completar el juego en alguna dificultad? *

Marca solo un óvalo.

☐ Sí

☐ No











11. ¿Qué tan divertido le pareció el juego? *

Marca solo un óvalo.

	1	2	3	4	5	
Muy aburrido	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy divertido

12. ¿Qué nota le pondría al juego? *

Marca solo un óvalo.

1	2	3	4	5	6	7	8	9	10
									

ANEXO C. ABSTRACT

The following pages have the purpose of summarizing the entire project for English speakers. It will have the same structure as the document, which is: an introduction, a short state of the art, the problem, the design of the video game, the result, the economic environment, the conclusions and the translation of the user's manual.

C.1. Introduction

The aim of this project is to explain the development of a Strategy Role Playing Game called Overworld Tactics.

The reasoning behind the decision of making a video game for a thesis has to do with the continuous growth of the gaming industry. Last year in Spain video games made more than 1359 millions of euros, that means a 16.9% increase comparing it to 2016 and making more money than movies and music together (597 and 232 millions of euros respectively).

The main objective of this work is to create a Strategy Role Playing Game or SRPG from now on, for desktop using the Java library LibGDX. Secondary objectives are to learn to use LibGDX to make games and to gain experience in the creation and design of video games.

C.2. State of the art

Before learning about Overworld Tactics it is important to know more about the history of video games and their actual state.

C.2.1. History of video games

The first video game (truly a proof of concept rather than a video game) was created by Alexander Douglas in 1952 and was called OXO, it was a version of Tic Tac Toe for the Electronic Delay Storage Automatic Calculator (EDSAC) computer, that is still operating in the University Mathematical Laboratory in Cambridge. You can see it in figure C.1.

Since the latter was a proof of concept, the actual first video game known is Spacewar!. Developed by Steve Russel, Martin Graetz, and Wayne Wiitanen at the Massachusetts Institute of Technology.

But it wasn't until Pong came out in 1976 published by Atari in its own machine that the first generation of consoles was born.

There are in total seven generations of consoles. Little details of each below these

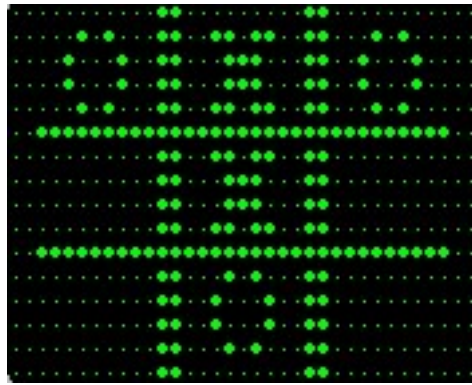


Fig. C.1. OXO video game [4]

lines.

1. Magnavox Odyssey, Atari Pong, and Coleco Telstar are the first consoles
2. Atari 2600 was the leader for this generation. More than 30 million units sold. SEGA's debut with SG 1000 and the Commodore 64 proved that it can be both a video game machine and a personal computer. Video game consoles thought to be dead.
3. Nintendo Famicom in Japan and Nintendo Entertainment System for the rest of the world revitalized the video game market with a standard for quality games and a limited number of licenses per developer a year. Super Mario Bros, Zelda, and Final Fantasy were created in this generation.
4. SEGA's Megadrive and Super Nintendo Entertainment System shared the market.
5. Three dimensions leap. Nintendo 64 and Sony's Playstation shared the market this time, leaving SEGA behind and the Apple and Bandai partnership with Apple Pip-pin that sold no more than fifty thousand units.
6. SEGA's last attempt with Dreamcast and the generation of the best selling console of all time (Figure C.2), the PlayStation 2, that came with a CD-ROM player. Also, the incursion of Microsoft (with Xbox) who saw in Sony's console a rival for their PCs.
7. Nintendo's comeback with the Wii. The generation of the motion controllers with the WiiMote, Kinect and PlayStation Move.

C.2.2. Video games nowadays

Right now we are in the eighth generation with the three main consoles being the Xbox One, PlayStation 4 and Nintendo Wii U. But this is a strange generation since there are revisions for the Xbox One, PS4, and the Wii U has been discontinued because of the lack of sells.

Pos	Platform	North America	Europe	Japan	Rest of World	Global
1	PlayStation 2 (PS2)	53.65	55.28	23.18	25.57	157.68
2	Nintendo DS (DS)	57.39	52.07	33.01	12.43	154.90
3	Game Boy (GB)	43.18	40.05	32.47	2.99	118.69
4	PlayStation (PS)	38.94	36.91	19.36	9.04	104.25
5	Wii (Wii)	45.51	33.88	12.77	9.48	101.64
6	PlayStation 3 (PS3)	29.42	34.55	10.47	12.46	86.90
7	Xbox 360 (X360)	49.11	25.87	1.66	9.16	85.80
8	Game Boy Advance (GBA)	40.39	21.31	16.96	2.85	81.51
9	PlayStation Portable (PSP)	21.41	24.14	20.01	15.26	80.82
10	PlayStation 4 (PS4)	26.92	32.03	6.76	14.21	79.92

Fig. C.2. Best selling consoles top 10 [10]

But now there is a new console. The Nintendo Switch that is the first console that can be played on the TV or like a portable and is selling pretty well, but there is no way to know if this is the start of the ninth generation or the second console from Nintendo in the same generation.

It is important also to highlight the importance of the smartphones and computers. Last year in Spain 23% of the gamer population played on consoles, versus the 21% that played in smartphones and the 22% that do the same in computers.

C.2.3. Indie games

Since the game developed for this thesis fits perfectly as an indie, it is important to talk about these games.

The start of indie games was in 2000 with the Scratchware Manifesto. Where a group of developer criticized the industry by saying that they were not taking risks with genres and doing innovative games. The industry was only moving for money.

C.3. The Problem

C.3.1. Strategy Role Playing Games

A good definition for an SRPG is that it is a game where the developing of the characters has an impact on the game and the player is the one making the decisions using strategy.

Some examples of SRPG's are Fire Emblem (Figure C.3) and Final Fantasy Tactics (Figure C.4). Overworld Tactics is inspired in both of these sagas, but more on Fire Emblem.

C.3.2. Tools

The tools used for the development of the game were LibGDX and Tiled. The first



Fig. C.3. Fire Emblem [19]



Fig. C.4. Final Fantasy Tactics [18]

one is the Java library where the game was developed and the second one is a level editor where the maps were created.

Other tools used are the following:

- Udacity
- IntelliJ IDEA
- Safari
- Safari Books Online
- Stack Overflow
- LucidChart
- Overleaf v2
- Github
- Mendeley
- Timing
- Hours
- Merlin Project Express

C.4. Game design

This section will be a brief of the truly Game Design Document.

C.4.1. Video game description

Overworld Tactics is an SRPG with two maps and three difficulties. It is aimed for a casual public that like strategy games.

The main objective is to clear both maps from the enemies.

C.4.2. Characters

There are three characters that can be controlled by the player. Those are the heroes. The rest of the characters are the enemies. All characters have three stats: attack, defense, and health.

Heroes are Eirika, the princess and heir to the throne; Christian, Eirika's little brother; and Airmanagild, the colonel.

There are two types of enemies: skeletons and limos. The first ones have a low defense but good attack. The Limos have great balanced stats.



Fig. C.5. Overworld Tactics' characters

All characters can be seen in the Figure C.5.

C.4.3. Story

This game does not have an explicit story. It starts with the level selection screen and then the gameplay begins. But it has an implicit story that will be explained in the following paragraph.

The kingdom where Eirika is the heir has been invaded by an army of monsters. So she, Christian and Airmanagild go on a quest to defeat them in the two regions of their homeplace.

C.4.4. Gameplay

Objectives

The objective of the game is to defeat all the monsters in the two maps.

Forms of interaction

The entire video game is played with the mouse. No keyboard controls required.

Game Mechanics

Overworld Tactics begins with the difficulty selection screen. Once the difficulty is chosen, it starts the proper gameplay. The first turn corresponds always to the player and she can do two actions with each hero: to move it or to attack an enemy.

To move a hero, the player needs to touch it and the Action Menu will pop, then touch in the Move button and the tiles where the unit can move will highlight on the map. Next step is to touch the tile where she wants to move the character. If the player doesn't want to move a character on a turn, she needs to click Move and then click out of the zone where the unit can move.

If a hero is adjacent to an enemy, it can attack it. To attack, the player needs to touch the unit, then on the Attack button and later on the enemy that she wants to inflict damage to.

Every time a hero is clicked, the Action Menu and the Information Window appears. The latter one has all the info of the hero's stats. Also, if the player touches an enemy in her turn, she can see the enemy's information as well as the zone where it can move.

The turn ends where all the units move and attack if they had the chance.

Since this is a Role Playing Game, the characters need to be developed. In this case, they develop two stats: attack and defense. When a unit attacks another one, the attack stat of the attacking unit goes up one point if the defending unit has an attack lower in five points than the attacking one. It goes up three points when the attacked unit has an attack higher than the attacking in five or more points and it only goes up one point in the other cases. Same happens with the defending unit and its defense stat.

The enemies' AI works by trying to find a hero that they can kill in one attack or the one with the lowest defense. First, they search in their own range of movement, and then globally.

Progression and challenge

Normally, in the SRPG games, the stats of the enemies don't grow at the same pace as the player's units. In this game they do. So every time the player is attacking an enemy, it gets stronger for the following turn. That is a thing that needs to be taken into account. Also, in the second screen the enemies are stronger, but so are the heroes.

There are also three difficulty levels, where the number of enemies and their stats change, but not the heroes' stats.

Losing

The game ends when all the heroes die. Either in the first screen or the second one. Game Over screen will show up and the player needs to restart the game from scratch.

C.4.5. Art Style

This game is a 2D top-down game with an overworld type map and tiles of 32x32 pixels.

C.4.6. Technical description

The game is compatible with PC that has Java Runtime Environment installed. The window size is fixed in 640x640 pixels and both maps will have 20x20 tiles with 32x32 pixels each. The frames per second are always 60.

C.5. Result

The result of this project is a Strategy Role Playing Game with two maps and three difficulty levels. That was the desired result. The figures from C.6 to C.7 show the game and some of its mechanics:

C.5.1. Versions

The game had several pre-releases before the final one. Here are the release notes of each one of them:



Fig. C.6. First map in Easy mode



Fig. C.7. After clicking on a hero



Fig. C.8. Selected hero's movement



Fig. C.9. Enemy's information



Fig. C.10. After an attack. Damage dealt is 3

Version 0.1.0

- First playable version. It has the Main Screen, one map, the action menu, the information window and units can move and attack.
- One enemy.
- Figure C.11.

Version 0.1.1

- Fixed a bug in the information window with the unit's name.

Version 0.2.0

- Damage dealt can be seen.
- Information window size is better.

Version 0.3.0

- First version with *overworld* map style. No more units climbing trees or the castle walls (like in figure C.12).



Fig. C.11. Version 0.1.0

- Three difficulties for the same map.
- Cell division visible in the map.

Version 0.4.0

- The chance to hit is now 70%. Before it was 100%.

Version 1.0.0

- First official release.
- A second map added for each difficulty (figure C.13).
- The chance to hit is now 85%.
- Labels centered in all screens.

Version 1.0.1

- Fixed a bug where the level didn't come back to the first one when you hit "Back to Main Menu" button.

To know more about tests made to the game and feedback received, please go to the chapter 5 of the main document.



Fig. C.12. Enemies “climbing” the castle walls



Fig. C.13. Segundo mapa del juego, en dificultad fácil

C.6. Economic environment

The cost of the entire project goes up to 7,638.03 €. Where 4,929.36 €, are human resources (more than 250 hours of work), 231.47 €, are material costs and the difference are indirect costs and Spain's Social Security. To see an extended version of the costs of the project, please refer to the main document where there are several tables with the detailed cost of each section.

There is no intention to win money with this project. As a matter of fact, the code will be released publicly in GitHub and the game published on Internet completely free with a Creative Commons license.

C.7. Conclusions

As said previously, the game industry makes more money than the movie and music together in Spain. So it is important to keep making video games because the public need to be entertained as much as it needs other things in life. It is not just an economic thing, but a health matter to be entertained and to entertain others.

Overworld Tactics is an SRPG with two screens and three levels of difficulty. Not long, it can be completed from 5 to 40 minutes depending on the difficulty and expertise of the player. It can be replayed several times to try to find new ways to defeat it.

It is also important to talk about the limitations and problems found while doing this project. The most important was the absence of time to work on the game on a daily basis, also the lack of experience with the tool chosen to develop the game (LibGDX) made the formation to the last longer than wanted.

Some improvements to be done in the future can be to add an explicit story to engage the players, more levels, more forms of interaction, new heroes, weapons and enemies. That is why the project will be kept in GitHub publicly, to let the community continue with it if there are people willing to.

C.8. User's Manual

C.8.1. Level Selection

There are three difficulties: easy, medium and hard. The heroes' stats don't change but the enemies' stats improve. Also, the enemies numbers vary in the hard difficulty. If this is your first time playing an SRPG, we strongly advise starting with easy.

C.8.2. How to play

The main objective of the game is to defeat all enemies in both maps.

In your turn, you have two actions you can do with each one of your characters: move or attack. Every time you touch your character, a menu pops out with these two buttons.

To attack an enemy, you have to be in an adjacent tile, so the normal procedure is to move near the enemy and attack it when you are next to it.

When an attack happens, a red number appears over the attacked unit, that's the damage done. If a black minus (-) appears, the attack missed. The possibilities are 85% to hit.

You only can move once, but if you attack first, then you can't move again. So think well what you should do. Also, each time you open the menu, you can see the information about your character, and to see the info about the enemies, you can touch them.

If you don't want to move or attack with a unit, you can leave it in the same spot by clicking the move button in the menu and then click outside of its movement area.

In their turn, the enemies will come close to you and attack, they follow the same rules as you and know the same about you that you know about them. They move thinking about your stats.

Your stats (and enemies') grow when you attack or defend, feel free to notice the change every time you open the information menu.